Restaurant Booking And Food Odering System

AMAN BAGADIYA ,0801CS211013,
B.tech 2nd year,SGSITS INDORE

January 30, 2023

1 AIM :

The aim of my project is to make a restaurant management and food odering system via which user can perform following functions :

A. USER FUNCTIONS

ORDER FOOD
BOOK TABLE AT RESTAURANT
CANCELL BOOKING
GET INFORMATION ABOUT RESTAURANT

. B.ADMIN FUNCTIONS

ADMIN AUTHENTICATION
SET/UPDATE GST RATES
GENERATE INVOICE
VIEW ALL INVOICES
VIEW ALL BOOKINGS

2 Following are the functions used in project:

2.1 Welcome_Page()

This function shows the user functionalities available for him also ask the user about the login choice and then depending upon the user selected choice calls the respective function.

2.2 Admin_Page()

This function allows the admin to perform following functions to update GST rate, to view all Invoices, to view all Bookings.

2.3 Admin_authentication()

This funtion is used to verify admin by the user entered password nd admin id with the already existing one.

2.4 User_Registration_Page()

This funtion takes the user details - name, phone number, address, email id as input and stores it into the structure.

2.5 options()

This function displays the options available to the user as follows 1.Get Information About Restaurant 2.Order Food:¿ 3.Book table:¿ 4.Cancel your Booking:¿ 5.Exit:¿ and gives the choice to select from the available option and calls the respective function.

2.6 place_order()

This function displays other 2 functions which are menu and city display.

2.7 city_display()

This is functions gives the user the choice of the city and ask him to select from the given choices.

2.8 City_choice(int city_choice)

This function intakes the perameter as city choice selected by the user and calls the respective city function.

2.9 INDORE()

This function shows the list of the available restaurants of Indore city.

2.10 BHOPAL()

This function shows the list of the available restaurants of Bhopal city,

2.11 MUMBAI()

This function shows the list of the available restaurants of Mumbai city.

2.12 INDORE_info(int restaurant_name)

This function shows the information to user about selected restaurant of Indore city.

2.13 BHOPAL_info(int restaurant_name)

This function shows the information to user about selected restaurant of Bhopal city.

2.14 MUMBAI_info(int restaurant_name)

This function shows the information to user about selected restaurant of Mumbai city.

2.15 void menu()

This function shows the menu and takes the item no. and its quantity as input and calls the bill function for bill generation.

2.16 void Bill()

Its function is to compute the bill and ask the user to add more items and also to call the invoice function to print invoice.

2.17 void Invoice()

Its function to display the invoice on screen in proper format and also to call save_details().

2.18 save_details()

This function is responsible for creating a file and saving the user details in it.

2.19 Booking()

This function is used to display the list of the vacant and occupied seats and also to book a table and display the successfull booking information.

2.20 save_booking_details()

This function is responsible for creating a file and saving the booking details in it.

2.21 Cancel_Booking()

This function is responsible for cancelling the booking by making that array element as 0.

2.22 void print_invoice()

This function is used to print the invoice from the file on screen.

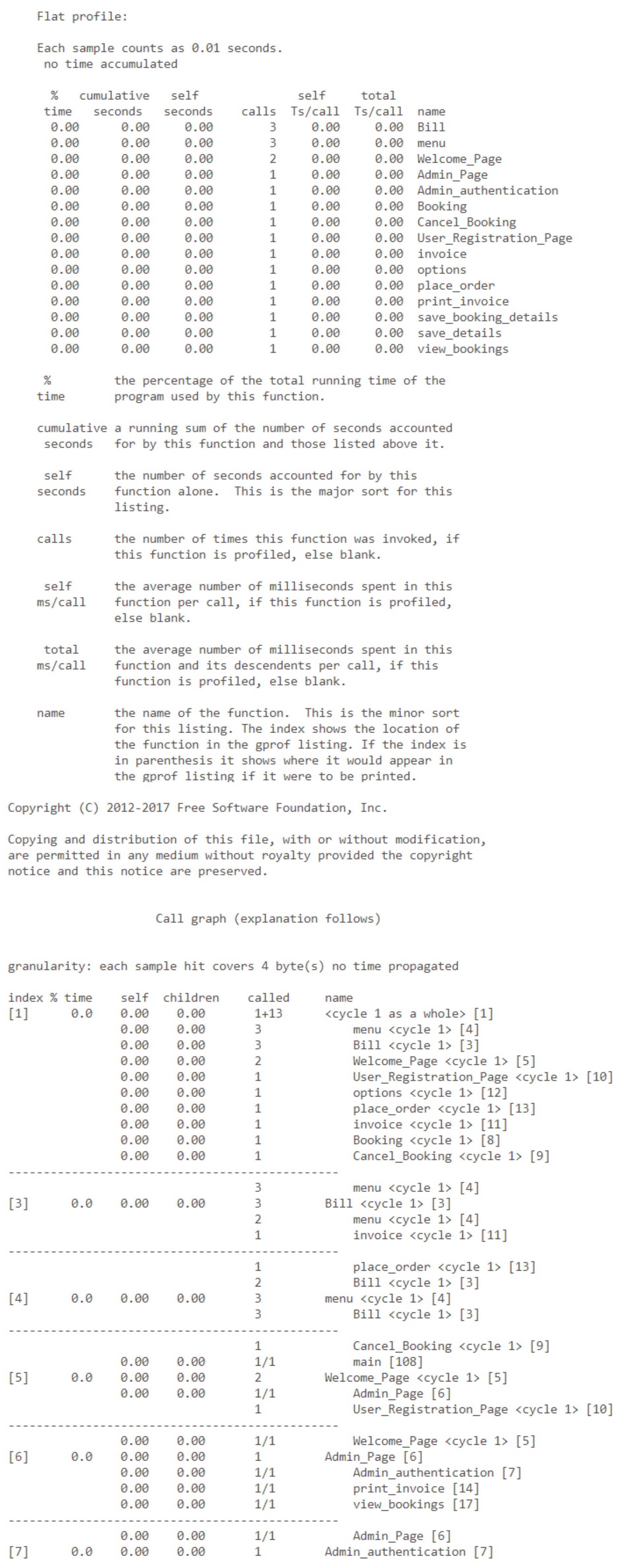
2.23 void view_bookings(

This function is used to print the bookings from the file on screen.

3 Description

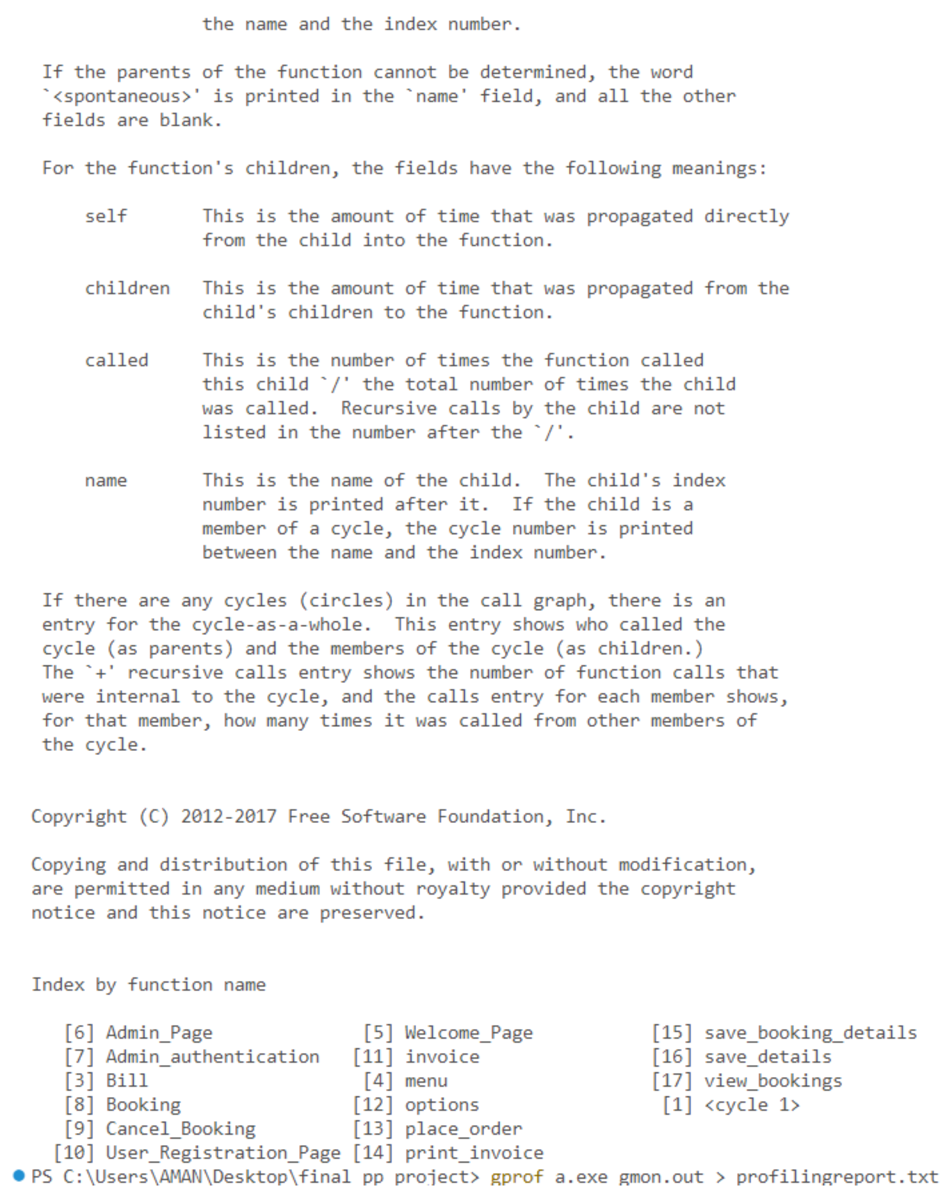
This mini project program is created with the objective to ease the food odering, restaurant booking and bill generation system it contains about 23 functions all of them perform different different work. It is created in 2 languages C and C++, Here i have majorly used the concept of array ,structures, file handeling,and loops.

4 Profilier report



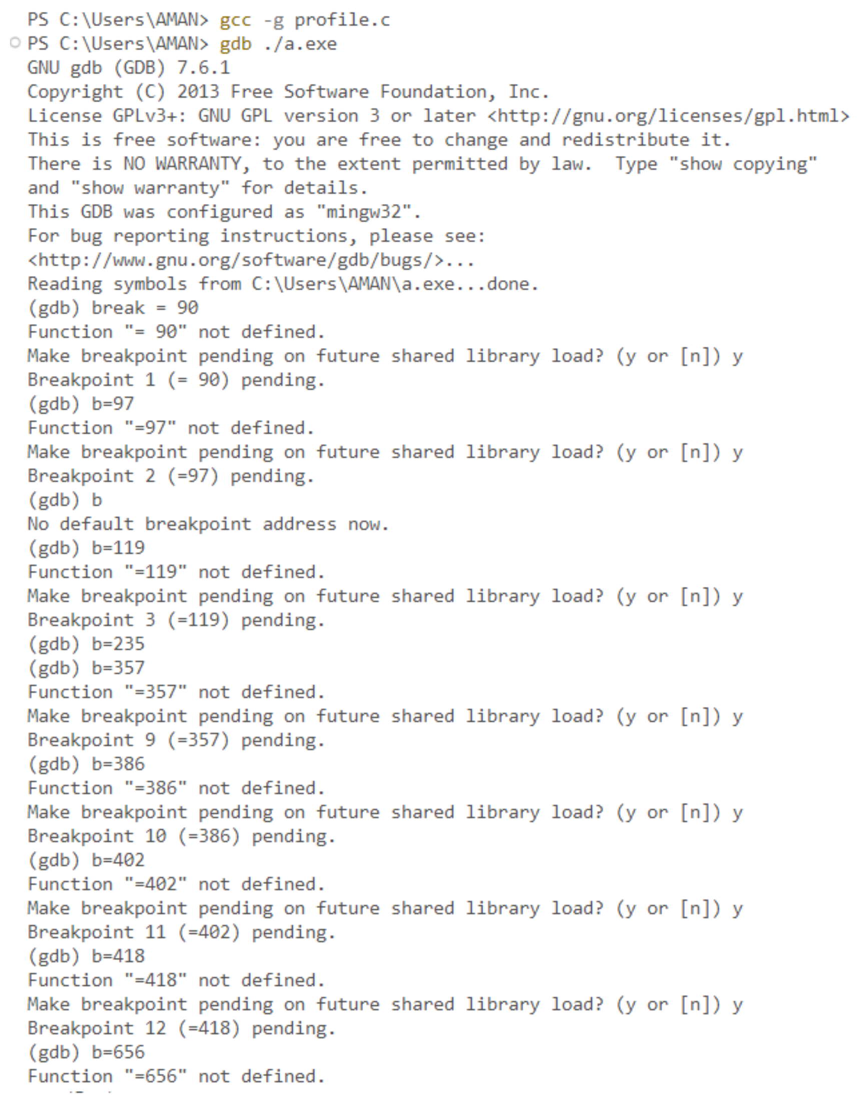
```
    Flat profile:

    Each sample counts as 0.01 seconds.
     no time accumulated

      %   cumulative   self              self     total
     time   seconds   seconds    calls  Ts/call  Ts/call  name
      0.00      0.00     0.00        3     0.00     0.00  Bill
      0.00      0.00     0.00        3     0.00     0.00  menu
      0.00      0.00     0.00        2     0.00     0.00  Welcome_Page
      0.00      0.00     0.00        1     0.00     0.00  Admin_Page
      0.00      0.00     0.00        1     0.00     0.00  Admin_authentication
      0.00      0.00     0.00        1     0.00     0.00  Booking
      0.00      0.00     0.00        1     0.00     0.00  Cancel_Booking
      0.00      0.00     0.00        1     0.00     0.00  User_Registration_Page
      0.00      0.00     0.00        1     0.00     0.00  invoice
      0.00      0.00     0.00        1     0.00     0.00  options
      0.00      0.00     0.00        1     0.00     0.00  place_order
      0.00      0.00     0.00        1     0.00     0.00  print_invoice
      0.00      0.00     0.00        1     0.00     0.00  save_booking_details
      0.00      0.00     0.00        1     0.00     0.00  save_details
      0.00      0.00     0.00        1     0.00     0.00  view_bookings

     %         the percentage of the total running time of the
    time       program used by this function.

    cumulative a running sum of the number of seconds accounted
     seconds   for by this function and those listed above it.

     self      the number of seconds accounted for by this
    seconds    function alone.  This is the major sort for this
               listing.

    calls      the number of times this function was invoked, if
               this function is profiled, else blank.

     self      the average number of milliseconds spent in this
    ms/call    function per call, if this function is profiled,
               else blank.

     total     the average number of milliseconds spent in this
    ms/call    function and its descendents per call, if this
               function is profiled, else blank.

    name       the name of the function.  This is the minor sort
               for this listing. The index shows the location of
               the function in the gprof listing. If the index is
               in parenthesis it shows where it would appear in
               the gprof listing if it were to be printed.

Copyright (C) 2012-2017 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.


                     Call graph (explanation follows)


granularity: each sample hit covers 4 byte(s) no time propagated

index % time    self  children    called     name
[1]      0.0    0.00    0.00      1+13      <cycle 1 as a whole> [1]
                0.00    0.00      3             menu <cycle 1> [4]
                0.00    0.00      3             Bill <cycle 1> [3]
                0.00    0.00      2             Welcome_Page <cycle 1> [5]
                0.00    0.00      1             User_Registration_Page <cycle 1> [10]
                0.00    0.00      1             options <cycle 1> [12]
                0.00    0.00      1             place_order <cycle 1> [13]
                0.00    0.00      1             invoice <cycle 1> [11]
                0.00    0.00      1             Booking <cycle 1> [8]
                0.00    0.00      1             Cancel_Booking <cycle 1> [9]
-----------------------------------------------
                                  3             menu <cycle 1> [4]
[3]      0.0    0.00    0.00      3         Bill <cycle 1> [3]
                                  2             menu <cycle 1> [4]
                                  1             invoice <cycle 1> [11]
-----------------------------------------------
                                  1             place_order <cycle 1> [13]
                                  2             Bill <cycle 1> [3]
[4]      0.0    0.00    0.00      3         menu <cycle 1> [4]
                                  3             Bill <cycle 1> [3]
-----------------------------------------------
                                  1             Cancel_Booking <cycle 1> [9]
                0.00    0.00      1/1           main [108]
[5]      0.0    0.00    0.00      2         Welcome_Page <cycle 1> [5]
                0.00    0.00      1/1           Admin_Page [6]
                                  1             User_Registration_Page <cycle 1> [10]
-----------------------------------------------
                0.00    0.00      1/1           Welcome_Page <cycle 1> [5]
[6]      0.0    0.00    0.00      1         Admin_Page [6]
                0.00    0.00      1/1           Admin_authentication [7]
                0.00    0.00      1/1           print_invoice [14]
                0.00    0.00      1/1           view_bookings [17]
-----------------------------------------------
                0.00    0.00      1/1           Admin_Page [6]
[7]      0.0    0.00    0.00      1         Admin_authentication [7]
```

```
-----------------------------------------------
                0.00    0.00       1/1           Admin_Page [6]
[7]      0.0    0.00    0.00       1         Admin_authentication [7]
-----------------------------------------------
                                   1             invoice <cycle 1> [11]
[8]      0.0    0.00    0.00       1         Booking <cycle 1> [8]
                0.00    0.00       1/1           save_booking_details [15]
                                   1             Cancel_Booking <cycle 1> [9]
-----------------------------------------------
                                   1             Booking <cycle 1> [8]
[9]      0.0    0.00    0.00       1         Cancel_Booking <cycle 1> [9]
                                   1             Welcome_Page <cycle 1> [5]
-----------------------------------------------
                                   1             Welcome_Page <cycle 1> [5]
[10]     0.0    0.00    0.00       1         User_Registration_Page <cycle 1> [10]
                                   1             options <cycle 1> [12]
-----------------------------------------------
                                   1             Bill <cycle 1> [3]
[11]     0.0    0.00    0.00       1         invoice <cycle 1> [11]
                0.00    0.00       1/1           save_details [16]
                                   1             Booking <cycle 1> [8]
-----------------------------------------------
                                   1             User_Registration_Page <cycle 1> [10]
[12]     0.0    0.00    0.00       1         options <cycle 1> [12]
                                   1             place_order <cycle 1> [13]
-----------------------------------------------
                                   1             options <cycle 1> [12]
[13]     0.0    0.00    0.00       1         place_order <cycle 1> [13]
                                   1             menu <cycle 1> [4]
-----------------------------------------------
                0.00    0.00       1/1           Admin_Page [6]
[14]     0.0    0.00    0.00       1         print_invoice [14]
-----------------------------------------------
                0.00    0.00       1/1           Booking <cycle 1> [8]
[15]     0.0    0.00    0.00       1         save_booking_details [15]
-----------------------------------------------
                0.00    0.00       1/1           invoice <cycle 1> [11]
[16]     0.0    0.00    0.00       1         save_details [16]
-----------------------------------------------
                0.00    0.00       1/1           Admin_Page [6]
[17]     0.0    0.00    0.00       1         view_bookings [17]
-----------------------------------------------

 This table describes the call tree of the program, and was sorted by
 the total amount of time spent in each function and its children.

 Each entry in this table consists of several lines.  The line with the
 index number at the left hand margin lists the current function.
 The lines above it list the functions that called this function,
    The lines above it list the functions that called this function,
    and the lines below it list the functions this one called.
    This line lists:
        index       A unique number given to each element of the table.
                    Index numbers are sorted numerically.
                    The index number is printed next to every function name so
                    it is easier to look up where the function is in the table.

        % time      This is the percentage of the `total' time that was spent
                    in this function and its children.  Note that due to
                    different viewpoints, functions excluded by options, etc,
                    these numbers will NOT add up to 100%.

        self        This is the total amount of time spent in this function.

        children    This is the total amount of time propagated into this
                    function by its children.

        called      This is the number of times the function was called.
                    If the function called itself recursively, the number
                    only includes non-recursive calls, and is followed by
                    a `+' and the number of recursive calls.

        name        The name of the current function.  The index number is
                    printed after it.  If the function is a member of a
                    cycle, the cycle number is printed between the
                    function's name and the index number.


    For the function's parents, the fields have the following meanings:

        self        This is the amount of time that was propagated directly
                    from the function into this parent.

        children    This is the amount of time that was propagated from
                    the function's children into this parent.

        called      This is the number of times this parent called the
                    function `/' the total number of times the function
                    was called.  Recursive calls to the function are not
                    included in the number after the `/'.

        name        This is the name of the parent.  The parent's index
                    number is printed after it.  If the parent is a
                    member of a cycle, the cycle number is printed between
                    the name and the index number.

    If the parents of the function cannot be determined, the word
    `<spontaneous>' is printed in the `name' field, and all the other
```

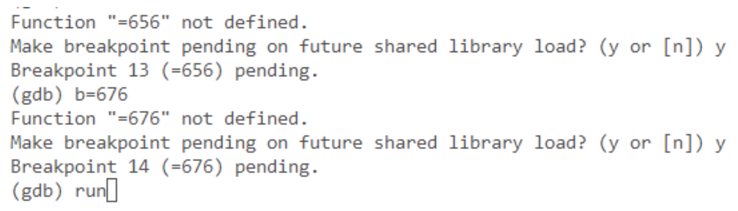


```
               the name and the index number.

 If the parents of the function cannot be determined, the word
 `<spontaneous>' is printed in the `name' field, and all the other
 fields are blank.

 For the function's children, the fields have the following meanings:

     self      This is the amount of time that was propagated directly
               from the child into the function.

     children  This is the amount of time that was propagated from the
               child's children to the function.

     called    This is the number of times the function called
               this child `/' the total number of times the child
               was called.  Recursive calls by the child are not
               listed in the number after the `/'.

     name      This is the name of the child.  The child's index
               number is printed after it.  If the child is a
               member of a cycle, the cycle number is printed
               between the name and the index number.

 If there are any cycles (circles) in the call graph, there is an
 entry for the cycle-as-a-whole.  This entry shows who called the
 cycle (as parents) and the members of the cycle (as children.)
 The `+' recursive calls entry shows the number of function calls that
 were internal to the cycle, and the calls entry for each member shows,
 for that member, how many times it was called from other members of
 the cycle.


Copyright (C) 2012-2017 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.


Index by function name

   [6] Admin_Page             [5] Welcome_Page          [15] save_booking_details
   [7] Admin_authentication  [11] invoice               [16] save_details
   [3] Bill                   [4] menu                  [17] view_bookings
   [8] Booking               [12] options                [1] <cycle 1>
   [9] Cancel_Booking        [13] place_order
  [10] User_Registration_Page [14] print_invoice
PS C:\Users\AMAN\Desktop\final pp project> gprof a.exe gmon.out > profilingreport.txt
```

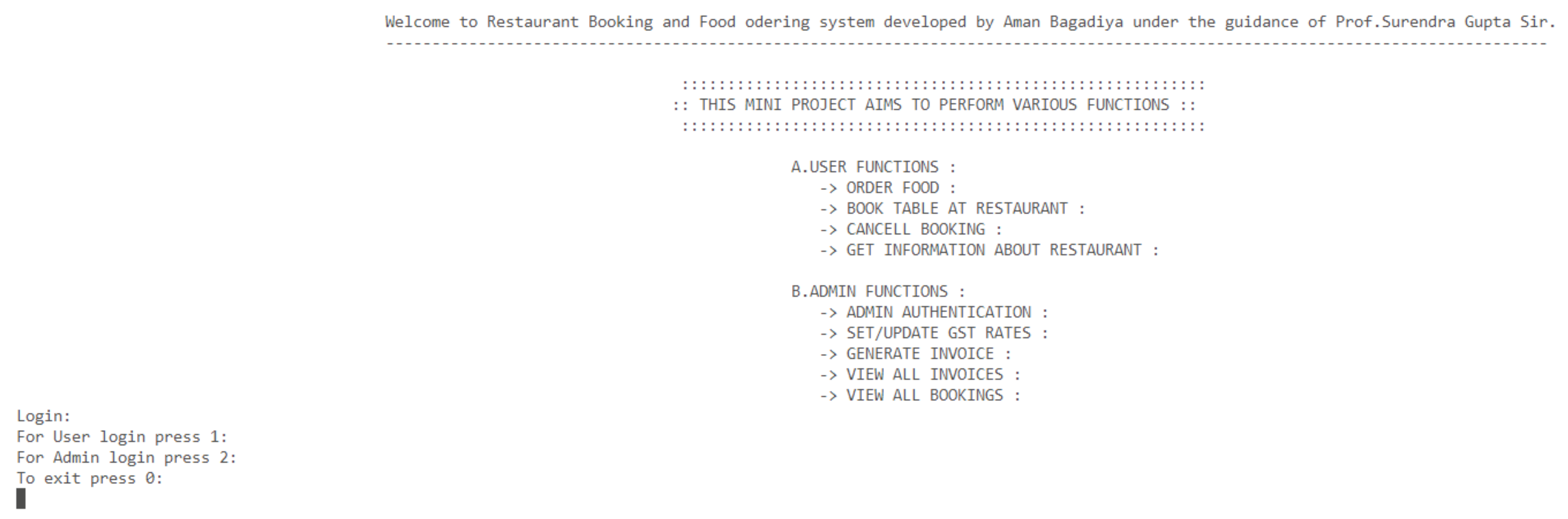
5 GBD Activities



```
PS C:\Users\AMAN> gcc -g profile.c
PS C:\Users\AMAN> gdb ./a.exe
GNU gdb (GDB) 7.6.1
Copyright (C) 2013 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "mingw32".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from C:\Users\AMAN\a.exe...done.
(gdb) break = 90
Function "= 90" not defined.
Make breakpoint pending on future shared library load? (y or [n]) y
Breakpoint 1 (= 90) pending.
(gdb) b=97
Function "=97" not defined.
Make breakpoint pending on future shared library load? (y or [n]) y
Breakpoint 2 (=97) pending.
(gdb) b
No default breakpoint address now.
(gdb) b=119
Function "=119" not defined.
Make breakpoint pending on future shared library load? (y or [n]) y
Breakpoint 3 (=119) pending.
(gdb) b=235
(gdb) b=357
Function "=357" not defined.
Make breakpoint pending on future shared library load? (y or [n]) y
Breakpoint 9 (=357) pending.
(gdb) b=386
Function "=386" not defined.
Make breakpoint pending on future shared library load? (y or [n]) y
Breakpoint 10 (=386) pending.
(gdb) b=402
Function "=402" not defined.
Make breakpoint pending on future shared library load? (y or [n]) y
Breakpoint 11 (=402) pending.
(gdb) b=418
Function "=418" not defined.
Make breakpoint pending on future shared library load? (y or [n]) y
Breakpoint 12 (=418) pending.
(gdb) b=656
Function "=656" not defined.
```



```
Function "=656" not defined.
Make breakpoint pending on future shared library load? (y or [n]) y
Breakpoint 13 (=656) pending.
(gdb) b=676
Function "=676" not defined.
Make breakpoint pending on future shared library load? (y or [n]) y
Breakpoint 14 (=676) pending.
(gdb) run
```



```
Welcome to Restaurant Booking and Food odering system developed by Aman Bagadiya under the guidance of Prof.Surendra Gupta Sir.
---------------------------------------------------------------------------------------------------------------------------

                              ::::::::::::::::::::::::::::::::::::::::::::::::::::::::
                              :: THIS MINI PROJECT AIMS TO PERFORM VARIOUS FUNCTIONS ::
                              ::::::::::::::::::::::::::::::::::::::::::::::::::::::::

                                        A.USER FUNCTIONS :
                                           -> ORDER FOOD :
                                           -> BOOK TABLE AT RESTAURANT :
                                           -> CANCELL BOOKING :
                                           -> GET INFORMATION ABOUT RESTAURANT :

                                        B.ADMIN FUNCTIONS :
                                           -> ADMIN AUTHENTICATION :
                                           -> SET/UPDATE GST RATES :
                                           -> GENERATE INVOICE :
                                           -> VIEW ALL INVOICES :
                                           -> VIEW ALL BOOKINGS :
Login:
For User login press 1:
For Admin login press 2:
To exit press 0:
```

6 Code with Comment in C Language

```
/*adding all the necessary header files*/
#include<stdio.h>
// #include<conio.h>
#include<stdlib.h>
#include<string.h>
/*Declaration of all the necessary functions */
void save\_booking\_details();
void Booking();
void Cancel\_Booking();
void menu();
void BHOPAL\_info();
void MUMBAI\_info();
void INDORE\_info();
void INDORE();
void City\_choice(int);
void city\_display();
void place\_order();
int  Admin\_authentication();
void Admin\_Page();
void User\_Registration\_Page ();
void Welcome\_Page();
void INDORE();
void BHOPAL();
void MUMBAI();
void options();
void save\_details();
void invoice();
void Bill();
void print\_invoice();
void view\_bookings();
/*Declaration of structure*/
struct User {
    char User\_Name[20];
    long long User\_contactdetails;
    char User\_address[50];
    char User\_emial\_id[30];
};
/*Declaration of variable of struct type*/
struct User user1;
/*Declaration of the other variables */
int Table\_no;
int gstRate;
int item;
int quantity;
float grandTotal;
int restaurant\_name;
int option;
int count=0;
/*Declaration of array to store data*/
char food\_name[50];
int item\_array[31] = {0};
char menu\_array[][35] = {"0","Paneer Angara", "Paneer Pasasnda", "Paneer Lababdar", "Paneer Tawa Masala", "Paneer Toofani", "
int quantity\_array[35] = {0};
int Table\_array[10] = {0};
int main (){
    printf ("Welcome to restaurant booking and food odering system developed by Aman Bagadiya");
    gstRate = 18; //default gst rates
    Welcome\_Page();  //calling of function ---> Welcome\_Page();
  }
```

```
//defining of the function Welcome\_Page
void Welcome\_Page(){
    system(”cls”); //this is the function which is defined in stlib.h and it is used to clear terminal screen
    printf (”\t\t\t\t\tWelcome to Restaurant Booking and Food odering system developed by Aman Bagadiya under the guidance of 
    printf (”\t\t\t\t\t------------------------------------------------------------------------------------------------
    printf(”\t\t\t\t\t                                   ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::\n”);
    printf (”\t\t\t\t\t                                  :: THIS MINI PROJECT AIMS TO PERFORM VARIOUS FUNCTIONS ::\n”);
    printf(”\t\t\t\t\t                                   ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::\n\n”);
    printf (”\t\t\t\t\t                                                A.USER FUNCTIONS :\n”);
    printf (”\t\t\t\t\t                                                   -> ORDER FOOD :\n”);
    printf (”\t\t\t\t\t                                                   -> BOOK TABLE AT RESTAURANT :\n”);
    printf (”\t\t\t\t\t                                                   -> CANCELL BOOKING :\n”);
    printf (”\t\t\t\t\t                                                   -> GET INFORMATION ABOUT RESTAURANT :\n\n”);
    printf (”\t\t\t\t\t                                                B.ADMIN FUNCTIONS :\n”);
    printf (”\t\t\t\t\t                                                   -> ADMIN AUTHENTICATION :\n”);
    printf (”\t\t\t\t\t                                                   -> SET/UPDATE GST RATES :\n”);
    printf (”\t\t\t\t\t                                                   -> GENERATE INVOICE :\n”);
    printf (”\t\t\t\t\t                                                   -> VIEW ALL INVOICES :\n”);
    printf (”\t\t\t\t\t                                                   -> VIEW ALL BOOKINGS :\n”);
    int Login\_option; //defining of the variable
    printf(”Login: \n”);
    printf(”For User login press 1: \n”);
    printf(”For Admin login press 2: \n”);
    printf(”To exit press 0: \n”);
    scanf (”%d”, &Login\_option);
    //HERE SWITCH CASE IS USED TO CALL THE CORRESPONDING OPTION DEPENDING ON THE USER CHOICE
     switch(Login\_option) {
        case 1: User\_Registration\_Page(); //calling of function ----> User\_Registration\_Page()
                break; // break statement is used to terminate switch statement execution,

        case 2: Admin\_Page(); //calling of function ----> Admin\_Page();
                break; // break statement is used to terminate switch statement execution,

        case 0: printf (”You have successfully exited from program \n\n ”);
                printf (”This project is made by Aman Bagadiya under the guidence of Mr Surendra Gupta Sir \n\n”);
                break; // break statement is used to terminate switch statement execution,

        default: printf(”You have entered an invalid choice... Exiting...\n \n”);
       exit(0); //exit(0) this function is used to successfully terminate and exit from the program
    }
}
//defining of the function Admin\_Page()
void Admin\_Page(){
     if (Admin\_authentication() == 1) {  // calling of the Admin\_authentication() function to verify the admin
     system(”cls”); //this is the function which is defined in stlib.h and it is used to clear terminal screen
     printf(”Current GST rate %d percent : \n”, gstRate);
     int select;//defining of the variable
     printf(”Press 1 to update GST rate:\n”);
     printf(”Press 2 to view all Invoices   :\n”);
     printf(”Press 3 to view all Bookings   :\n”);
     printf(”To exit press 0: \n”);
     scanf(”%d”,&select);
     //HERE SWITCH CASE IS USED TO CALL THE CORRESPONDING OPTION DEPENDING ON THE ADMIN CHOICE
        switch(select) {
            case 1:
            printf(”Enter New GST rates (in percentage): \n”);
            scanf(”%d”,&gstRate);
            case 2:
            print\_invoice(); //calling of function ---->  print\_invoice()
            case 3:
            view\_bookings(); //calling of function ----> view\_bookings()
        }
    }
}
//defining of the function Admin\_authentication()
int Admin\_authentication(){
    system(”cls”); //This is the function which is defined in stlib.h and it is used to clear terminal screen
    int admin\_check = 1;//defining of the variable
    char loginID[] =”Aman”;
    char password[] =”Aman”;
    char user\_entered\_password[11]; //Declaration of the array user\_entered\_password to store login id
    char user\_entered\_login[10];    //Declaration of the array user\_entered\_login to store password

   printf(”****************\t Welcome to Admin Login Page \t*****************\n\n”);
   printf(”Kindly enter your credentials below :\n\n”);
   printf(”Enter Admin login ID: ”);
   scanf(”%s”, user\_entered\_login);
   printf(”Enter  Admin password: ”);
   scanf(”%s”, user\_entered\_password);
   if(strcmp(loginID , user\_entered\_login) != 0) {  //Using strcmp function to compare 2 strings loginID and user\_entered\_
        admin\_check = 0;
   }
   if(strcmp(password, user\_entered\_password) != 0) { //Using strcmp function to compare 2 strings password and user\_entered
    admin\_check = 0;
   }
   if(admin\_check == 0) {
    printf("sorry You Have Entered Wrong credentials.\n");
    printf("Try Again\n");
    return 0;
    }else {
    return 1;  // if admin is successfully verified it will return 1
   }
}

void User\_Registration\_Page(){
  //defining of the function User\_Registration\_Page()
    system("cls"); //this is the function which is defined in stlib.h and it is used to clear terminal screen
    printf("Enter Your Name\n");
    fflush(stdin); //it is used to clear the buffer
    scanf("%[^\n]s", &user1.User\_Name);
    printf("Enter Your Phone Number\n");
    scanf("%lld", &user1.User\_contactdetails);
    printf("Enter Your Email-id\n");
    fflush(stdin);  //it is used to clear the buffer
    scanf("%[^\n]s", &user1.User\_emial\_id);
    printf("Enter Address\n");
    fflush(stdin);  //it is used to clear the buffer
    scanf("%[^\n]s", &user1.User\_address);
    printf("Your details registered successfully\n");
    options();  //calling of function ---> options
}
void options(){
    do{
    system("cls"); //this is the function which is defined in stlib.h and it is used to clear terminal screen
    printf("\n                          ::::::::::::::::::::::::::::::::::::::::::::::::::::::::");
    printf("\n                          ::         !!!!!!!!!!!!welcome!!!!!!!!!!!!          ::");
    printf("\n                          ::                                                  ::");
    printf("\n                          ::    ~> Please select from below options:          ::");
    printf("\n                          ::                                                  ::");
    printf("\n                          ::       1.Get Information About Restaurant :>      ::");
    printf("\n                          ::       2.Order Food:>                             ::");
    printf("\n                          ::       3.Book table:>                             ::");
    printf("\n                          ::       4.Cancel your Booking:>                    ::");
    printf("\n                          ::       5.Exit:>                                   ::");
    printf("\n                          ::::::::::::::::::::::::::::::::::::::::::::::::::::::::\n\n");
    printf("PLEASE SELECT FROM THE ABOVE OPTIONS:\n");
    scanf("%d", &option);
     //HERE SWITCH CASE IS USED TO CALL THE CORRESPONDING OPTION DEPENDING ON THE ADMIN CHOICE
    switch (option){
            case 1: city\_display();
                    break; // break statement is used to terminate switch statement execution,
            case 2 :
                    place\_order();
                    break; // break statement is used to terminate switch statement execution,
            case 3 : {
                    city\_display();
                    Booking();
                    }
            case 4 : {
                    city\_display();
                    Cancel\_Booking();
                    }
            case 5 : {
                    printf("\nExiting...\n");
                    exit(0); //exit(0) this function is used to successfully terminate and exit from the program
                  }
            default :{
                printf("Please Enter A Valid Choice");
            }
        }
      }while(1);
}
void place\_order(){
        city\_display();
        menu();
}
void city\_display(){
   system("cls"); //this is the function which is defined in stlib.h and it is used to clear terminal screen
   printf("                          ::::::::::::::::::::::::::::::::::::::::::::::::::::\n");
   printf("                          ::                                                ::\n");
   printf("                          ::    ~  Please select your city from below list: ::\n");
   printf("                          ::                                                ::\n");
   printf("                          ::     >           1.INDORE                       ::\n");
   printf("                          ::     >>          2.BHOPAL                       ::\n");
   printf("                          ::     >>>         3.MUMBAI                       ::\n");
   printf("                          ::                                                ::\n");
   printf("                          ::                                                ::\n");
   printf("                          ::                                                ::\n");
   printf("                          ::::::::::::::::::::::::::::::::::::::::::::::::::::\n\n");
   //getchar();
```

```
    printf("Enter your choice by entering the serial number of city : ");
    int city\_choice;
    scanf("%d",&city\_choice);
    City\_choice(city\_choice); //calling of function ---> City\_choice(city\_choice)
    }
void City\_choice(int city\_choice){
    switch (city\_choice){
          //HERE SWITCH CASE IS USED TO CALL THE CORRESPONDING OPTION DEPENDING ON THE ADMIN CHOICE
          case 1 : {
              INDORE();
              break; // break statement is used to terminate switch statement execution,
          }
          case 2 : {
              BHOPAL();
              break; // break statement is used to terminate switch statement execution,
          }
          case 3 : {
              MUMBAI();
              break; // break statement is used to terminate switch statement execution,
          }
      }
}
void INDORE(){
    system("cls"); //this is the function which is defined in stlib.h and it is used to clear terminal screen
    printf("                        ::::::::::::::::::::::::::::::::::::::::::::::::::::::\n");
    printf("                        ::    ~> Please select restaurant below list:       ::\n");
    printf("                        ::                                                  ::\n");
    printf("                        ::       Name of Restaurant        Ratings          ::\n");
    printf("                        ::                                                  ::\n");
    printf("                        ::       1.Kebabsville             **               ::\n");
    printf("                        ::       2.Ginger Ganesha          ***              ::\n");
    printf("                        ::       3.Shree Chotiwala         *                ::\n");
    printf("                        ::       4.Village                 **               ::\n");
    printf("                        ::       5.Vidorra                 *                ::\n");
    printf("                        ::                                                  ::\n");
    printf("                        ::::::::::::::::::::::::::::::::::::::::::::::::::::::\n\n");
    //getchar();
    printf("Enter your choice by entering the serial number of restaurant : \n");
    int restaurant\_name;
    scanf("%d",&restaurant\_name);
    if (option == 1)
    INDORE\_info(restaurant\_name); //calling of function ---> INDORE\_info
    printf("\n\n");
}
void BHOPAL(){
    system("cls"); //this is the function which is defined in stlib.h and it is used to clear terminal screen
    printf("                        ::::::::::::::::::::::::::::::::::::::::::::::::::::::\n");
    printf("                        ::    ~> Please select restaurant below list:       ::\n");
    printf("                        ::                                                  ::\n");
    printf("                        ::       Name of Restaurant        Ratings          ::\n");
    printf("                        ::                                                  ::\n");
    printf("                        ::       1.Manohar                 **               ::\n");
    printf("                        ::       2.Cafe Chokolade          **               ::\n");
    printf("                        ::       3.Greek Food & Beyond     *                ::\n");
    printf("                        ::       4.Bapu Ki Kutia           **               ::\n");
    printf("                        ::       5.Al-Beik                 ***              ::\n");
    printf("                        ::                                                  ::\n");
    printf("                        ::::::::::::::::::::::::::::::::::::::::::::::::::::::\n\n");
    //getchar();
    printf("Enter your choice by entering the serial number of restaurant : \n");
    int restaurant\_name;
    scanf("%d",&restaurant\_name);
    if (option == 1)
    BHOPAL\_info(restaurant\_name);  //calling of function ---> BHOPAL\_info
    printf("\n\n");

}
void MUMBAI(){
     system("cls"); //this is the function which is defined in stlib.h and it is used to clear terminal screen
    printf("                        ::::::::::::::::::::::::::::::::::::::::::::::::::::::\n");
    printf("                        ::    ~> Please select restaurant below list:       ::\n");
    printf("                        ::                                                  ::\n");
    printf("                        ::       Name of Restaurant        Ratings          ::\n");
    printf("                        ::                                                  ::\n");
    printf("                        ::       1.Bayroute Restaurant     **               ::\n");
    printf("                        ::       2. Hakkasan               ***              ::\n");
    printf("                        ::       3.Dome Intercontinental   *                ::\n");
    printf("                        ::       4.Yauatcha Restaurant     **               ::\n");
    printf("                        ::       5.Pali Village cafe       *                ::\n");
    printf("                        ::                                                  ::\n");
    printf("                        ::::::::::::::::::::::::::::::::::::::::::::::::::::::\n\n");
    //getchar();
    printf("Enter your choice by entering the serial number of restaurant : \n");

    scanf("%d",&restaurant\_name);
    if (option == 1)
    MUMBAI\_info(restaurant\_name);  //calling of function ---->   MUMBAI\_info
    printf("\n\n");

}
void INDORE\_info(int restaurant\_name){
    system("cls"); //this is the function which is defined in stlib.h and it is used to clear terminal screen
    switch (restaurant\_name){
            case 1 : {
                printf("LOCATION : Sayaji Indore, H/1 , Scheme No.54, Vijay Nagar, Indore, Madhya Pradesh - 452010\n\nPHONE :07
                break; // break statement is used to terminate switch statement execution,
            }
            case 2 : {
                printf("LOCATION LG-12,13, DM Tower, 21/1, Lala Banarasilal Dawar Marg, Race Course Road, Near Zanjeerwala Chov
                break; // break statement is used to terminate switch statement execution,
            }
            case 3 : {
                printf("LOCATION: 8 B, Raunak Plaza, Opposite Nath Mandir, South Tukoganj, Indore, Madhya Pradesh 452001, India
One of the most popular eating joints in the city, Shree Chotiwala serves you with delicious veg Indian food. The wide variety
                break; // break statement is used to terminate switch statement execution,
            }
            case 4 : {
                printf("LOCATION: 5th Floor, Central Mall, RNT Marg, Indore, Madhya Pradesh 452001, India\n\nPHONE: +91-731-40
                break; // break statement is used to terminate switch statement execution,
            }
            case 5 : {
                printf("LOCATION : New Palasia, Indore\n\nCONTACT DETAILS:12345678 \n\nMORE INFORMATION :CUISINES Indian, Asian
                break; // break statement is used to terminate switch statement execution,
            }
    }
    printf("\nPress any key to continue...");
    //getch(); //Here //getch will take any character input and then this function will be terminated and the next line of the
}
void BHOPAL\_info(int restaurant\_name){
    system("cls"); //this is the function which is defined in stlib.h and it is used to clear terminal screen
      //HERE SWITCH CASE IS USED TO CALL THE CORRESPONDING OPTION DEPENDING ON THE USER CHOICE
     switch (restaurant\_name){
            case 1 : {
                printf("132, Zone 1, Maharana Pratap Nagar, Bhopal\nStreet Food, South Indian, Fast Food, Desserts, North India
            }
            case 2 : {
                printf("\n206, Zone 2, Near Arya Bhavan, Maharana Pratap Nagar, Bhopal,\n India Cafe, Bakery \n11 AM to 11 PM \
\n");
                break; // break statement is used to terminate switch statement execution,
            }
            case 3 : {
                printf("Third Floor, DB City Mall, Maharana Pratap Nagar, Bhopal, \nIndia Greek, Mediterranean \n12 Noon to 10:
                break; // break statement is used to terminate switch statement execution,
            }

            case 4 : {
               printf(" 123, Jyoti Shopping Centre, Behind Jyothi Talkies Zone-I, Maharana Pratap Nagar,Bhopal, Madhya Pradesh
               break; // break statement is used to terminate switch statement execution,
            }

            case 5 : {
              printf("Regiment Rd,Badabagh, Shahjahanabad,Bhopal, Madhya Pradesh 462001\n Snacks \n1:30 PM to 10 PM\n INR 300\r
              break; // break statement is used to terminate switch statement execution,
            }
        }
    printf("\nPress any key to continue...");
    //getch(); //Here //getch will take any character input and then this function will be terminated and the next line of the
}
void MUMBAI\_info(int restaurant\_name){
    system("cls"); //this is the function which is defined in stlib.h and it is used to clear terminal screen
      //HERE SWITCH CASE IS USED TO CALL THE CORRESPONDING OPTION DEPENDING ON THE USER CHOICE
     switch (restaurant\_name){
            case 1 : {
                printf("Location: Bayroute Juhu, 14, Silver Beach Estate Opposite Juhu Post Office, AB Nair Road, Juhu, Mumbai.
                break; // break statement is used to terminate switch statement execution,
            }
            case 2 : {
                printf("Location: Hakkasan, 2nd Floor, Krystal Building, Waterfield Road, Bandra West, Mumbai. \nFoods to try:
                break; // break statement is used to terminate switch statement execution,
            }
            case 3 : {
                printf("Location: Dome Intercontinental 135, Marine Drive, Mumbai, \nFoods to try: Afghani Chicken, Brownie and
                break; // break statement is used to terminate switch statement execution,
            }
            case 4 : {
                printf("Location: Yauatcha, Raheja Towers, Bandra Kurla Complex, Bandra East, Mumbai\n Foods to try: Crispy Pra
                break; // break statement is used to terminate switch statement execution,
            }
            case 5 : {
                printf("Location: Pali Village Cafe, 602, Ambedkar Road, Pali Naka, Pali Hill, Bandra West, Mumbai\n Foods to t
Risotto, Salads, Panna Cotta, Pizzas are worth a try. \nPrice for two: INR 3000\n");
                break; // break statement is used to terminate switch statement execution,
            }
    }
    printf("\nPress any key to continue...");
    //getch(); //Here //getch will take any character input and then this function will be terminated and the next line of the
}
void menu(){
  //defining of the function menu()
   printf("\n :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::");
   printf("\n\n  ITEMS.................................................................PRICE \n1  Paneer Angara..............
Paneer Pasasnda ......................................................210 \n3  Paneer Lababdar................................
Paneer Tawa Masala....................................................215 \n5  Paneer Toofani ................................
Paneer Handi..........................................................200 \n7  Paneer Kadai...................................
Paneer Tikka Masala...................................................195 \n9  Paneer Butter Masala...........................
   printf("\n:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::\n\n");
   //getchar();
   printf("ENTER THE S.NO. OF THE ITEM FROM THE ABOVE LIST :\n");
   scanf("%d",&item);
   printf("ENTER THE QUANTITY: \n");
   scanf("%d",&quantity);
   Bill();
}
void Bill(){
  //defining of the function Bill()
    item\_array[count] = item;
    quantity\_array[count] = quantity;
    count++;
    float price;
    //HERE SWITCH CASE IS USED TO CALL THE CORRESPONDING OPTION DEPENDING ON THE ITEM CHOOSEN BY THE USER
    switch (item)
    {
    case 1:
        price = 220 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 2:
        price = 210 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 3:
       price = 215 * quantity;
       break; // break statement is used to terminate switch statement execution,
    case 4:
       price = 215 * quantity;
       break; // break statement is used to terminate switch statement execution,
    case 5:
       price = 220 * quantity;
       break; // break statement is used to terminate switch statement execution,
    case 6:
       price = 200 * quantity;
       break; // break statement is used to terminate switch statement execution,
    case 7:
       price = 200 * quantity;
       break; // break statement is used to terminate switch statement execution,
    case 8:
       price = 195 * quantity;
       break; // break statement is used to terminate switch statement execution,
    case 9:
       price = 195 * quantity;
       break; // break statement is used to terminate switch statement execution,
    case 10:
       price = 188 * quantity;
       break; // break statement is used to terminate switch statement execution,
    case 11:
       price = 88 * quantity;
       break; // break statement is used to terminate switch statement execution,
    case 12:
       price = 78 * quantity;
       break; // break statement is used to terminate switch statement execution,
    case 13:
       price = 98 * quantity;
       break; // break statement is used to terminate switch statement execution,
    case 14:
       price = 48 * quantity;
       break; // break statement is used to terminate switch statement execution,
    case 15:
       price = 230 * quantity;
       break; // break statement is used to terminate switch statement execution,
    case 16:
       price = 195 * quantity;
       break; // break statement is used to terminate switch statement execution,
    case 17:
       price = 205 * quantity;
       break; // break statement is used to terminate switch statement execution,
    case 18:
       price = 180 * quantity;
```

```
        break; // break statement is used to terminate switch statement execution,
    case 19:
        price = 180 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 20:
        price = 195 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 21:
        price = 130 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 22:
        price = 130 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 23:
        price = 85 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 24:
        price = 100 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 25:
        price = 90 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 26:
        price = 140 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 27:
        price = 130 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 28:
        price = 140 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 29:
        price = 130 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 30:
        price = 170 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 31:
        price = 55 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 32:
         price = 55 * quantity;
         break; // break statement is used to terminate switch statement execution,
    case 33:
         price = 45 * quantity;
         break; // break statement is used to terminate switch statement execution,
    case 34:
         price = 45 * quantity;
         break; // break statement is used to terminate switch statement execution,
    }

    float gst = (price * gstRate) / 100;
    float total = price + gst;
    grandTotal += total;
    int add;
    printf("DO YOU WANT TO ADD MORE ITEMS TO YOUR CART YES = 1 / NO = 0 or press any other key to return to main menu :");
    scanf("%d", &add);
    switch(add){
        case 0:{
            invoice();
            break; // break statement is used to terminate switch statement execution,
        }
        case 1: {
            menu();
            break; // break statement is used to terminate switch statement execution,
        }
    }
}
//defining of the function invoice()
void invoice(){
        system("cls"); //this is the function which is defined in stlib.h and it is used to clear terminal screen
        printf("************************************** :) YOUR INVOICE (: **************************************\n\n\n");

        printf("                              Customer Name     :  %s\n\n", user1.User\_Name);
        printf("                              Customer Address  :  %s\n\n", user1.User\_address);
        printf("                              Contact  Details  :  %lu\n\n", user1.User\_contactdetails);
        printf("                              Email-id          :  %s\n\n", user1.User\_emial\_id);
        printf("                              Item name                  Quantity       \n\n");
        for (int i = 0; i < count; i++)
        {
        printf("                              %s ----------> %d-only\n\n ",menu\_array[item\_array[i]],quantity\_array[i]);
        }
        printf("------------------------------------------------------------------------------------------------------------\n\n");
        printf("                               Total              :   %3f Rs/- Only\n\n\n\n", grandTotal);
        printf("*********************************** Thank you! Do visit again :) 1*********************************** ");
    printf("\nPress any key to continue...");
    //getch(); //Here //getch will take any character input and then this function will be terminated and the next line of the
    save\_details();  //calling of function ——> save\_details()
    exit(0); //exit(0) this function is used to successfully terminate and exit from the program
}
void save\_details(){
    FILE * fILE\_Pointer; //creation of file pointer
    fILE\_Pointer = fopen("invoice.txt", "a+"); //opening the file invoice.txt in append plus mode and assigning it to file poi
    /* Write data to file */
    fprintf(fILE\_Pointer,"*************************************** :) YOUR INVOICE (: ****************************************

        fprintf(fILE\_Pointer,"                              Customer Name    :  %s\n\n", user1.User\_Name);
        fprintf(fILE\_Pointer,"                              Customer Address :  %s\n\n", user1.User\_address);
        fprintf(fILE\_Pointer,"                              Contact  Details :  %lu\n\n", user1.User\_contactdetails);
        fprintf(fILE\_Pointer,"                              Email-id         :  %s\n\n", user1.User\_emial\_id);
        fprintf(fILE\_Pointer,"                              Item name                Quantity      \n\n");
        for (int i = 0; i < count; i++)
        {
        fprintf(fILE\_Pointer,"                              %s ————————> %d-only\n\n ",menu\_array[item\_array[i]],quantity\_array[
        }
        fprintf(fILE\_Pointer,"——————————————————————————————————————————————————————————————————————————————————————————————————————
        fprintf(fILE\_Pointer,"                               Total              :   %3f Rs/- Only\n\n\n\n", grandTotal);
    /* Close file to save file data */
        fclose(fILE\_Pointer);
}
void Booking(){
    system("cls"); //this is the function which is defined in stlib.h and it is used to clear terminal screen
    printf("Here is the list of vacant and occupied tables\n\n");
    /*using the for loop to traverse through the array and check each value of element of array and print booked if it is 1 and
    for (int k = 0; k < 10; k++){
        if (Table\_array[k]==1){
            printf("Table no. %d is already Booked\n", k + 1);
        }
        else {
            printf("Table no. %d is vacant\n", k + 1);
        }
    }

    printf("Enter the table number of the table you want to book\n");
    scanf("%d", &Table\_no);
    Table\_array[Table\_no+1] == 1;/*assigning 1 to [Table\_no+1] of the array to mark it as booked */
    system("cls"); //this is the function which is defined in stlib.h and it is used to clear terminal screen
    printf("                        :::::::::::::::::::::::::::::::::::::::::::::::::::::\n");
    printf("                         Customer Name    :  %s\n\n", user1.User\_Name);
    printf("                         Contact  Details :  %lu\n\n", user1.User\_contactdetails);
    printf("                         Email-id         :  %s\n\n", user1.User\_emial\_id);
    printf("                         Congratulations table number %d successfully booked\n",Table\_no);
    printf("                        :::::::::::::::::::::::::::::::::::::::::::::::::::::\n");
    save\_booking\_details();  //calling of function ——> save\_booking\_details()
    exit(0); //exit(0) this function is used to successfully terminate and exit from the program
}
void save\_booking\_details(){
    FILE * FILE\_Pointer;  //creation of file pointer
    FILE\_Pointer = fopen("booking.txt", "a+"); //opening the file booking.txt in append plus mode and assigning it to file poi
    /* Write data to file */
    fprintf(FILE\_Pointer,"                        :::::::::::::::::::::::::::::::::::::::::::::::::::::\n");
    fprintf(FILE\_Pointer,"                         Customer Name    :  %s\n\n", user1.User\_Name);
    fprintf(FILE\_Pointer,"                         Contact  Details :  %lu\n\n", user1.User\_contactdetails);
    fprintf(FILE\_Pointer,"                         Email-id         :  %s\n\n", user1.User\_emial\_id);
    fprintf(FILE\_Pointer,"                         Congratulations table number %d successfully booked\n",Table\_no);
    fprintf(FILE\_Pointer,"                        :::::::::::::::::::::::::::::::::::::::::::::::::::::\n");
    /* Close file to save file data */
    fclose(FILE\_Pointer);
}
void Cancel\_Booking(){
    int cancel;
    system("cls"); //this is the function which is defined in stlib.h and it is used to clear terminal screen
    printf("Enter table number to cancel your booking \n\n");
    scanf("%d", &cancel);
    Table\_array[cancel+1] == 0; /*assigning 0 to [cancel+1] of the array to mark it as vacant */
    system("cls"); //this is the function which is defined in stlib.h and it is used to clear terminal screen
    printf("                        :::::::::::::::::::::::::::::::::::::::::::::::::::::\n\n");
    printf("                         Customer Name    :  %s\n\n", user1.User\_Name);
    printf("                         Contact  Details :  %lu\n\n", user1.User\_contactdetails);
    printf("                         Email-id         :  %s\n\n", user1.User\_emial\_id);
    printf("                         Your booking for table number %d successfully canceled\n\n",cancel);
    printf("                        :::::::::::::::::::::::::::::::::::::::::::::::::::::\n");
    exit(0); //exit(0) this function is used to successfully terminate and exit from the program

}
void print\_invoice(){
   FILE *file\_pointer; //creation of file pointer
   file\_pointer = fopen("invoice.txt", "r+"); //opening the file invoice.txt in read plus mode and assigning it to file pointe
   char c;   //declaration of char c to read data from the file
```

```
    c = fgetc(file\_pointer);  //fgetc is used to fetch data from the file
    /*while loop is used it will print data until the eof occurs*/
    while (c != EOF)
    {
        printf(”%c”, c);
        c = fgetc(file\_pointer);
     }
     /* Close file to save file data */
     fclose(file\_pointer);
     exit(0); //exit(0) this function is used to successfully terminate and exit from the program
}
void view\_bookings(){
      FILE *file1\_pointer; //creation of file pointer
      file1\_pointer = fopen(”booking.txt”, ”r+”); //opening the file booking.txt in read plus mode and assigning it to file poi
      char c;  //declaration of char c to read data from the file
      c = fgetc(file1\_pointer); //fgetc is used to fetch data from the file
     /*while loop is used it will print data until the eof occurs*/
      while (c != EOF)
     {
        printf(”%c”, c);
        c = fgetc(file1\_pointer);
     }
     fclose(file1\_pointer);  /* Close file to save file data */
     exit(0); //exit(0) this function is used to successfully terminate and exit from the program
}
```

7 Code with Comment in C++ Language

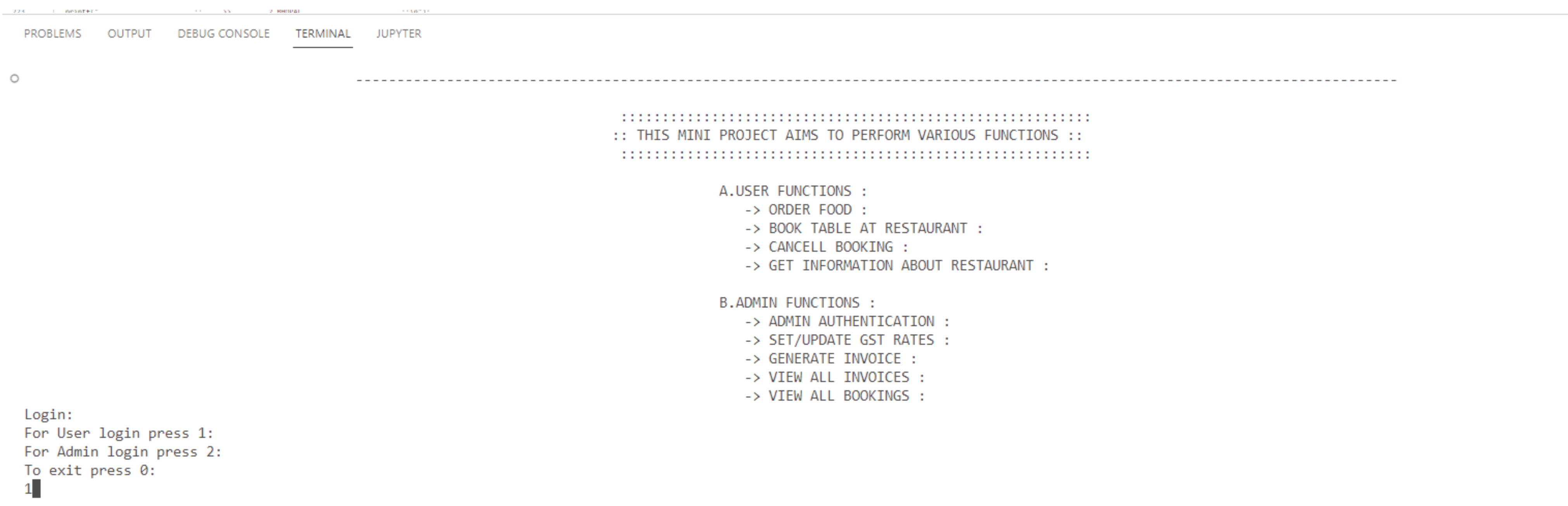
```
/*adding all the necessary header files*/
#include <iostream>
#include <cstring>

/*Declaration of class*/
class User
{
private:
    char User_Name[20];
    long long User_contactdetails;
    char User_address[50];
    char User_emial_id[30];

public:
    /*Declaration of the other variables */
    int Table_no;
    int gstRate;
    int item;
    int quantity;
    float grandTotal;
    int restaurant_name;
    int option;
    int count = 0;

    /*Declaration of all the necessary functions */
    void save_booking_details();
    void Booking();
    void Cancel_Booking();
    void menu();
    void BHOPAL_info(int restaurant_name);
    void MUMBAI_info(int restaurant_name);
    void INDORE_info(int restaurant_name);
    void City_choice(int);
    void city_display();
    void place_order();
    int Admin_authentication();
    void Admin_Page();
    void User_Registration_Page();
    void INDORE();
    void BHOPAL();
    void MUMBAI();
    void options();
    void save_details();
    void invoice();
    void Bill();
    void print_invoice();
    void Welcome_Page();
    void view_bookings();
} user1; /*Declaration of variable of class type*/

/*Declaration of array to store data*/
char food_name[50];
int item_array[31] = {0};
int quantity_array[35] = {0};
int Table_array[10] = {0};
char menu_array[][35] = {"Nothing Selected", "Paneer Angara", "Paneer Pasasnda", "Paneer Lababdar", "Paneer Tawa Masala", "Pan

int main()
{
    std::cout << "Welcome to restaurant booking and food odering system developed by Aman Bagadiya";
    user1.gstRate = 18;   // default gst rates
    user1.Welcome_Page(); // calling of function ——> Welcome_Page();
}

// defining of the function Welcome_Page
void User::Welcome_Page()
{
    system("cls");    // this is the function which is defined in stlib.h and it is used to clear terminal screen
    int Login_option; // defining of the variable
    std :: cout<<"\t\t\t\t\tWelcome to Restaurant Booking and Food odering system developed by Aman Bagadiya under the guidance
   std :: cout<<"\t\t\t\t\t——————————————————————————————————————————————————————————————————————————————————————
   std :: cout<<"\t\t\t\t\t                              ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::\n";
   std :: cout<<"\t\t\t\t\t                              :: THIS MINI PROJECT AIMS TO PERFORM VARIOUS FUNCTIONS ::\n";
   std :: cout<<"\t\t\t\t\t                              ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::\n\n";
   std :: cout<<"\t\t\t\t\t                                          A.USER FUNCTIONS :\n";
   std :: cout<<"\t\t\t\t\t                                             —> ORDER FOOD :\n";
   std :: cout<<"\t\t\t\t\t                                             —> BOOK TABLE AT RESTAURANT :\n";
   std :: cout<<"\t\t\t\t\t                                             —> CANCELL BOOKING :\n";
   std :: cout<<"\t\t\t\t\t                                             —> GET INFORMATION ABOUT RESTAURANT :\n\n";
   std :: cout<<"\t\t\t\t\t                                          B.ADMIN FUNCTIONS :\n";
   std :: cout<<"\t\t\t\t\t                                             —> ADMIN AUTHENTICATION :\n";
```

```
    std :: cout<<"\t\t\t\t\t                                                   -> SET/UPDATE GST RATES :\n";
    std :: cout<<"\t\t\t\t\t                                                   -> GENERATE INVOICE :\n";
    std :: cout<<"\t\t\t\t\t                                                   -> VIEW ALL INVOICES :\n";
    std :: cout<<"\t\t\t\t\t                                                   -> VIEW ALL BOOKINGS :\n";
    std ::cout << "Login: \n";
    std ::cout << "For User login press 1: \n";
    std ::cout << "For Admin login press 2: \n";
    std ::cout << "To exit press 0: \n";
    std ::cin >> Login_option;
    // HERE SWITCH CASE IS USED TO CALL THE CORRESPONDING OPTION DEPENDING ON THE USER CHOICE
    switch (Login_option)
    {
    case 1:
        User_Registration_Page(); // calling of function ----> User_Registration_Page()
        break;                    // break statement is used to terminate switch statement execution,

    case 2:
        Admin_Page(); // calling of function ----> Admin_Page();
        break;        // break statement is used to terminate switch statement execution,

    case 0:
        std ::cout << "You have successfully exited from program \n\n ";
        std ::cout << "This project is made by Aman Bagadiya under the guidence of Mr Surendra Gupta Sir \n\n";
        break; // break statement is used to terminate switch statement execution,

    default:
        std ::cout << "You have entered an invalid choice...Exiting...\n \n";
        exit(0); // exit(0) this function is used to successfully terminate and exit from the program
    }
}

// defining of the function Admin_Page()
void User ::Admin_Page()
{
    if (Admin_authentication() == 1)
    {   // calling of the Admin_authentication() function to verify the admin
        system("cls"); // this is the function which is defined in stlib.h and it is used to clear terminal screen
        std ::cout << "Current GST rate " << gstRate << " percent : \n";
        int select; // defining of the variable
        std ::cout << "Press 1 to update GST rate:\n";
        std ::cout << "Press 2 to view all Invoices   :\n";
        std ::cout << "Press 3 to view all Bookings   :\n";
        std ::cout << "To exit press 0: \n";
        std ::cin >> select;
        // HERE SWITCH CASE IS USED TO CALL THE CORRESPONDING OPTION DEPENDING ON THE ADMIN CHOICE
        switch (select)
        {
        case 1:
            std ::cout << "Enter New GST rates (in percentage): \n";
            std ::cin >> gstRate;
            Welcome_Page();

        case 2:
            print_invoice(); // calling of function ---->  print_invoice()
        case 3:
            view_bookings(); // calling of function ----> view_bookings()
        }
    }
}

// defining of the function Admin_authentication()
int User ::Admin_authentication()
{
    system("cls");        // This is the function which is defined in stlib.h and it is used to clear terminal screen
    int admin_check = 1; // defining of the variable
    char loginID[] = "Aman";
    char password[] = "Aman";
    char user_entered_password[11]; // Declaration of the array user_entered_password to store login id
    char user_entered_login[10];    // Declaration of the array user_entered_login to store password

    std ::cout << "****************\t Welcome to Admin Login Page \t*****************\n\n";
    std ::cout << "Kindly enter your credentials below :\n\n";
    std ::cout << "Enter Admin login ID: ";
    std ::cin >> user_entered_login;
    std ::cout << "Enter Admin password: ";
    std ::cin >> user_entered_password;
    if (strcmp(loginID, user_entered_login) != 0)
    { // Using strcmp function to compare 2 strings loginID and user_entered_login
        admin_check = 0;
    }
    if (strcmp(password, user_entered_password) != 0)
    { // Using strcmp function to compare 2 strings password and user_entered_password
        admin_check = 0;
    }
    if (admin_check == 0)
    {
        std ::cout << "Sorry You Have Entered Wrong Credentials.\n";
        std ::cout << "Try Again\n";
        return 0;
    }
    else
    {
        return 1; // if admin is successfully verified it will return 1
    }
}

void User ::User_Registration_Page()
{
    // defining of the function User_Registration_Page()
    system("cls"); // this is the function which is defined in stlib.h and it is used to clear terminal screen
    std ::cout << "Enter Your Name\n";
    fflush(stdin); // it is used to clear the buffer
    gets(user1.User_Name);
    std ::cout << "Enter Your Phone Number\n";
    std ::cin >> user1.User_contactdetails;
    std ::cout << "Enter Your Email-id\n";
    fflush(stdin); // it is used to clear the buffer
    gets(user1.User_emial_id);
    std ::cout << "Enter Address\n";
    fflush(stdin); // it is used to clear the buffer
    gets(user1.User_address);
    std ::cout << "Your details registered successfully\n";
    options(); // calling of function ---> options
}

void User ::options()
{
    do
    {
        system("cls"); // this is the function which is defined in stlib.h and it is used to clear terminal screen
        std ::cout << "\n                         ::::::::::::::::::::::::::::::::::::::::::::::::::::::::";
        std ::cout << "\n                         ::          !!!!!!!!!!!!welcome!!!!!!!!!!!!             ::";
        std ::cout << "\n                         ::                                                      ::";
        std ::cout << "\n                         ::    ~> Please select from below options:              ::";
        std ::cout << "\n                         ::                                                      ::";
        std ::cout << "\n                         ::       1.Get Information About Restaurant :>          ::";
        std ::cout << "\n                         ::       2.Order Food:>                                 ::";
        std ::cout << "\n                         ::       3.Book table:>                                 ::";
        std ::cout << "\n                         ::       4.Cancel your Booking:>                        ::";
        std ::cout << "\n                         ::       5.Exit:>                                       ::";
        std ::cout << "\n                         ::::::::::::::::::::::::::::::::::::::::::::::::::::::::\n\n";
        std ::cout << "PLEASE SELECT FROM THE ABOVE OPTIONS:\n";
        std ::cin >> option;
        // HERE SWITCH CASE IS USED TO CALL THE CORRESPONDING OPTION DEPENDING ON THE ADMIN CHOICE
        switch (option)
        {
        case 1:
            city_display();
            break; // break statement is used to terminate switch statement execution,
        case 2:
            place_order();
            break; // break statement is used to terminate switch statement execution,
        case 3:
        {
            city_display();
            Booking();
        }
        case 4:
        {
            city_display();
            Cancel_Booking();
        }
        case 5:
        {
            std ::cout << "\nExiting...\n";
            exit(0); // exit(0) this function is used to successfully terminate and exit from the program
        }
        default:
        {
            std ::cout << "Please Enter A Valid Choice";
        }
        }
    } while (1);
}

void User ::place_order()
{
    city_display();
    menu();
}
```

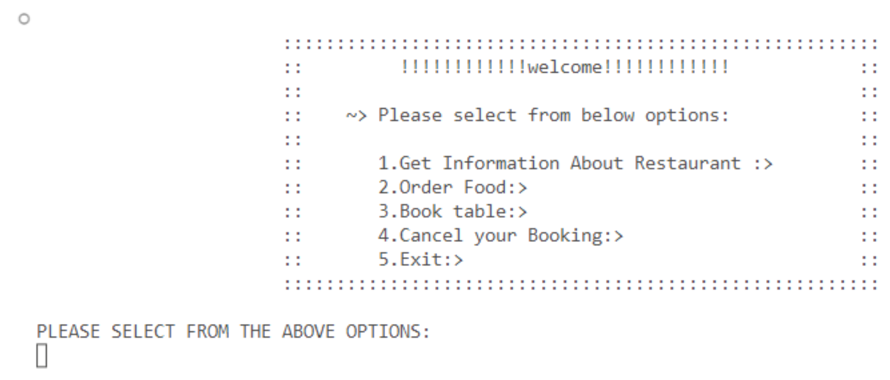
```
void User ::city_display()
{
    system("cls"); // this is the function which is defined in stlib.h and it is used to clear terminal screen
    std ::cout << "                        :::::::::::::::::::::::::::::::::::::::::::::::::::\n";
    std ::cout << "                        ::                                               ::\n";
    std ::cout << "                        ::    ~  Please select your city from below list: ::\n";
    std ::cout << "                        ::                                               ::\n";
    std ::cout << "                        ::     >          1.INDORE                       ::\n";
    std ::cout << "                        ::     >>         2.BHOPAL                       ::\n";
    std ::cout << "                        ::     >>>        3.MUMBAI                       ::\n";
    std ::cout << "                        ::                                               ::\n";
    std ::cout << "                        ::                                               ::\n";
    std ::cout << "                        ::                                               ::\n";
    std ::cout << "                        :::::::::::::::::::::::::::::::::::::::::::::::::::\n\n";
    getchar();
    std ::cout << "Enter your choice by entering the serial number of city : ";
    int city_choice;
    std ::cin >> city_choice;
    City_choice(city_choice); // calling of function ---> City_choice(city_choice)
}

void User ::City_choice(int city_choice)
{
    switch (city_choice)
    {
    // HERE SWITCH CASE IS USED TO CALL THE CORRESPONDING OPTION DEPENDING ON THE ADMIN CHOICE
    case 1:
    {
        INDORE();
        break; // break statement is used to terminate switch statement execution,
    }
    case 2:
    {
        BHOPAL();
        break; // break statement is used to terminate switch statement execution,
    }
    case 3:
    {
        MUMBAI();
        break; // break statement is used to terminate switch statement execution,
    }
    }
}

void User ::INDORE()
{
    system("cls"); // this is the function which is defined in stlib.h and it is used to clear terminal screen
    std ::cout << "                        :::::::::::::::::::::::::::::::::::::::::::::::::::::\n";
    std ::cout << "                        ::    ~> Please select restaurant below list:      ::\n";
    std ::cout << "                        ::                                                 ::\n";
    std ::cout << "                        ::       Name of Restaurant        Ratings         ::\n";
    std ::cout << "                        ::                                                 ::\n";
    std ::cout << "                        ::       1.Kebabsville             **              ::\n";
    std ::cout << "                        ::       2.Ginger Ganesha          ***             ::\n";
    std ::cout << "                        ::       3.Shree Chotiwala         *               ::\n";
    std ::cout << "                        ::       4.Village                 **              ::\n";
    std ::cout << "                        ::       5.Vidorra                 *               ::\n";
    std ::cout << "                        ::                                                 ::\n";
    std ::cout << "                        :::::::::::::::::::::::::::::::::::::::::::::::::::::\n\n";
    getchar();
    std ::cout << "Enter your choice by entering the serial number of restaurant : \n";
    int restaurant_name;
    std ::cin >> restaurant_name;
    if (option == 1)
        INDORE_info(restaurant_name); // calling of function ---> INDORE_info
    std ::cout << "\n\n";
}

void User ::BHOPAL()
{
    system("cls"); // this is the function which is defined in stlib.h and it is used to clear terminal screen
    std ::cout << "                        :::::::::::::::::::::::::::::::::::::::::::::::::::::\n";
    std ::cout << "                        ::    ~> Please select restaurant below list:      ::\n";
    std ::cout << "                        ::                                                 ::\n";
    std ::cout << "                        ::       Name of Restaurant        Ratings         ::\n";
    std ::cout << "                        ::                                                 ::\n";
    std ::cout << "                        ::       1.Manohar                 **              ::\n";
    std ::cout << "                        ::       2.Cafe Chokolade          **              ::\n";
    std ::cout << "                        ::       3.Greek Food & Beyond     *               ::\n";
    std ::cout << "                        ::       4.Bapu Ki Kutia           **              ::\n";
    std ::cout << "                        ::       5.Al-Beik                 ***             ::\n";
    std ::cout << "                        ::                                                 ::\n";
    std ::cout << "                        :::::::::::::::::::::::::::::::::::::::::::::::::::::\n\n";
    getchar();
    std ::cout << ”Enter your choice by entering the serial number of restaurant : \n”;
    int restaurant_name;
    std ::cin >> restaurant_name;
    if (option == 1)
        BHOPAL_info(restaurant_name); // calling of function ——> BHOPAL_info
    std ::cout << ”\n\n”;
}

void User ::MUMBAI()
{
    system(”cls”); // this is the function which is defined in stlib.h and it is used to clear terminal screen
    std ::cout << ”                        ::::::::::::::::::::::::::::::::::::::::::::::::::::::\n”;
    std ::cout << ”                        ::    ~> Please select restaurant below list:       ::\n”;
    std ::cout << ”                        ::                                                  ::\n”;
    std ::cout << ”                        ::       Name of Restaurant        Ratings          ::\n”;
    std ::cout << ”                        ::                                                  ::\n”;
    std ::cout << ”                        ::       1.Bayroute Restaurant     **               ::\n”;
    std ::cout << ”                        ::       2. Hakkasan               ***              ::\n”;
    std ::cout << ”                        ::       3.Dome Intercontinental   *                ::\n”;
    std ::cout << ”                        ::       4.Yauatcha Restaurant     **               ::\n”;
    std ::cout << ”                        ::       5.Pali Village cafe       *                ::\n”;
    std ::cout << ”                        ::                                                  ::\n”;
    std ::cout << ”                        ::::::::::::::::::::::::::::::::::::::::::::::::::::::\n\n”;
    getchar();
    std ::cout << ”Enter your choice by entering the serial number of restaurant : \n”;

    std ::cin >> restaurant_name;
    if (option == 1)
        MUMBAI_info(restaurant_name); // calling of function ——>   MUMBAI_info
    std ::cout << ”\n\n”;
}

void User ::INDORE_info(int restaurant_name)
{
    system(”cls”); // this is the function which is defined in stlib.h and it is used to clear terminal screen
    switch (restaurant_name)
    {
    case 1:
    {
        std ::cout << ”LOCATION : Sayaji Indore, H/1 , Scheme No.54, Vijay Nagar, Indore, Madhya Pradesh – 452010\n\nPHONE :073
        exit(0); // break statement is used to terminate switch statement execution,
    }
    case 2:
    {
        std ::cout << ”LOCATION LG−12,13, DM Tower, 21/1, Lala Banarasilal Dawar Marg, Race Course Road, Near Zanjeerwala Chow
        exit(0); // break statement is used to terminate switch statement execution,
    }
    case 3:
    {
        std ::cout << ”LOCATION: 8 B, Raunak Plaza, Opposite Nath Mandir, South Tukoganj, Indore, Madhya Pradesh 452001, India\
One of the most popular eating joints in the city, Shree Chotiwala serves you with delicious veg Indian food. The wide variety
        exit(0); // break statement is used to terminate switch statement execution,
    }
    case 4:
    {
        std ::cout << ”LOCATION: 5th Floor, Central Mall, RNT Marg, Indore, Madhya Pradesh 452001, India\n\nPHONE: +91−731−400
        exit(0); // break statement is used to terminate switch statement execution,
    }
    case 5:
    {
        std ::cout << ”LOCATION : New Palasia, Indore\n\nCONTACT DETAILS:12345678 \n\nMORE INFORMATION :CUISINES Indian, Asian
        exit(0); // break statement is used to terminate switch statement execution,
    }
    }
    std ::cout << ”\nPress any key to continue...”;
}

void User ::BHOPAL_info(int restaurant_name)
{
    system(”cls”); // this is the function which is defined in stlib.h and it is used to clear terminal screen
                   // HERE SWITCH CASE IS USED TO CALL THE CORRESPONDING OPTION DEPENDING ON THE USER CHOICE
    switch (restaurant_name)
    {
    case 1:
    {
        std ::cout << ”132, Zone 1, Maharana Pratap Nagar, Bhopal\nStreet Food, South Indian, Fast Food, Desserts, North Indian
    }
    case 2:
    {
        std ::cout << ”\n206, Zone 2, Near Arya Bhavan, Maharana Pratap Nagar, Bhopal,\n India Cafe, Bakery \n11 AM to 11 PM \
\n”;
        break; // break statement is used to terminate switch statement execution,
    }
    case 3:
    {
        std ::cout << "Third Floor, DB City Mall, Maharana Pratap Nagar, Bhopal, \nIndia Greek, Mediterranean \n12 Noon to 10:3
        break; // break statement is used to terminate switch statement execution,
    }

    case 4:
    {
        std ::cout << " 123, Jyoti Shopping Centre, Behind Jyothi Talkies Zone−I, Maharana Pratap Nagar,Bhopal, Madhya Pradesh
        break; // break statement is used to terminate switch statement execution,
    }

    case 5:
    {
        std ::cout << "Regiment Rd,Badabagh, Shahjahanabad,Bhopal, Madhya Pradesh 462001\n Snacks \n1:30 PM to 10 PM\n INR 300\
        break; // break statement is used to terminate switch statement execution,
    }
    }
    std ::cout << "\nPress any key to continue...";

}

void User ::MUMBAI_info(int restaurant_name)
{
    system("cls"); // this is the function which is defined in stlib.h and it is used to clear terminal screen
                   // HERE SWITCH CASE IS USED TO CALL THE CORRESPONDING OPTION DEPENDING ON THE USER CHOICE
    switch (restaurant_name)
    {
    case 1:
    {
        std ::cout << "Location: Bayroute Juhu, 14, Silver Beach Estate Opposite Juhu Post Office, AB Nair Road, Juhu, Mumbai.\
        break; // break statement is used to terminate switch statement execution,
    }
    case 2:
    {
        std ::cout << "Location: Hakkasan, 2nd Floor, Krystal Building, Waterfield Road, Bandra West, Mumbai. \nFoods to try: I
        break; // break statement is used to terminate switch statement execution,
    }
    case 3:
    {
        std ::cout << "Location: Dome Intercontinental 135, Marine Drive, Mumbai, \nFoods to try: Afghani Chicken, Brownie and
        break; // break statement is used to terminate switch statement execution,
    }
    case 4:
    {
        std ::cout << "Location: Yauatcha, Raheja Towers, Bandra Kurla Complex, Bandra East, Mumbai\n Foods to try: Crispy Prav
        break; // break statement is used to terminate switch statement execution,
    }
    case 5:
    {
        std ::cout << "Location: Pali Village Cafe, 602, Ambedkar Road, Pali Naka, Pali Hill, Bandra West, Mumbai\n Foods to tr
Risotto, Salads, Panna Cotta, Pizzas are worth a try. \nPrice for two: INR 3000\n";
        break; // break statement is used to terminate switch statement execution,
    }
    }
    std ::cout << "\nPress any key to continue...";
}

void User ::menu()
{
    // defining of the function menu()
    std ::cout << "\n :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::";
    std ::cout << "\n\n  ITEMS...............................................................PRICE \n1  Paneer Angara.....
Paneer Pasasnda .......................................................210 \n3  Paneer Lababdar................................
Paneer Tawa Masala.....................................................215 \n5  Paneer Toofani ................................
Paneer Handi...........................................................200 \n7  Paneer Kadai...................................
Paneer Tikka Masala....................................................195 \n9  Paneer Butter Masala...........................
    std ::cout << "\n::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::\n\n";
    getchar();
    std ::cout << "ENTER THE S.NO. OF THE ITEM FROM THE ABOVE LIST :\n";
    std ::cin >> item;
    std ::cout << "ENTER THE QUANTITY: \n";
    std ::cin >> quantity;
    Bill();
}

void User ::Bill()
{
    // defining of the function Bill()
    item_array[count] = item;
    quantity_array[count] = quantity;
    count++;
    int price;
    // HERE SWITCH CASE IS USED TO CALL THE CORRESPONDING OPTION DEPENDING ON THE ITEM CHOOSEN BY THE USER
    switch (item)
    {
    case 1:
        price = 220 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 2:
        price = 210 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 3:
        price = 215 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 4:
        price = 215 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 5:
        price = 220 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 6:
        price = 200 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 7:
        price = 200 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 8:
        price = 195 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 9:
        price = 195 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 10:
        price = 188 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 11:
        price = 88 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 12:
        price = 78 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 13:
        price = 98 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 14:
        price = 48 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 15:
        price = 230 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 16:
        price = 195 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 17:
        price = 205 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 18:
        price = 180 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 19:
        price = 180 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 20:
        price = 195 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 21:
        price = 130 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 22:
        price = 130 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 23:
        price = 85 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 24:
        price = 100 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 25:
        price = 90 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 26:
        price = 140 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 27:
        price = 130 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 28:
        price = 140 * quantity;
        break; // break statement is used to terminate switch statement execution,
    case 29:
        price = 130 * quantity;
```

```
            break; // break statement is used to terminate switch statement execution,
        case 30:
            price = 170 * quantity;
            break; // break statement is used to terminate switch statement execution,
        case 31:
            price = 55 * quantity;
            break; // break statement is used to terminate switch statement execution,
        case 32:
            price = 55 * quantity;
            break; // break statement is used to terminate switch statement execution,
        case 33:
            price = 45 * quantity;
            break; // break statement is used to terminate switch statement execution,
        case 34:
            price = 45 * quantity;
            break; // break statement is used to terminate switch statement execution,
        }

        float gst = (price * gstRate) / 100;
        float total = price + gst;
        grandTotal += total;
        int add;
        std::cout << "DO YOU WANT TO ADD MORE ITEMS TO YOUR CART YES = 1 / NO = 0 or press any other key to return to main menu :"
        std::cin >> add;
        switch (add)
        {
        case 0:
        {
            invoice();
            break; // break statement is used to terminate switch statement execution,
        }
        case 1:
        {
            menu();
            break; // break statement is used to terminate switch statement execution,
        }
        }
    }

    // defining of the function invoice()
    void User::invoice()
    {
        system("cls"); // this is the function which is defined in stlib.h and it is used to clear terminal screen
        std::cout << "*************************************** :) YOUR INVOICE (: ***************************************\n\n\n";

        std::cout << "                          Customer Name    :  " << user1.User_Name << " \n\n";
        std::cout << "                          Customer Address :  " << user1.User_address << " \n\n";
        std::cout << "                          Contact  Details :  " << user1.User_contactdetails << " \n\n";
        std::cout << "                          Email-id         :  " << user1.User_emial_id << " \n\n";
        std::cout << "                          Item name                Quantity       \n\n";
        for (int i = 0; i < count; i++)
        {
            std::cout << "                          "<< menu_array[item_array[i]] <<" ----------> "<<  quantity_array[i] <<"-only\n\n
        }
        std::cout << "--------------------------------------------------------------------------------------------------------------\n\n";

        std::cout << "                            Total             :   "<< grandTotal << " Rs/- Only\n\n\n\n";

        std::cout << "********************************** Thank you! Do visit again :) 1********************************** ";
        std::cout << "\nPress any key to continue...";
        getchar(); //Here getch will take any character input and then this function will be terminated and the next line of the c

        save_details(); // calling of function ----> save_details()
        exit(0);        // exit(0) this function is used to successfully terminate and exit from the program
    }

    void User::save_details()
    {
        FILE *fILE_Pointer;                          // creation of file pointer
        fILE_Pointer = fopen("invoice.txt", "a+"); // opening the file invoice.txt in append plus mode and assigning it to file poi
        /* Write data to file */
        fprintf(fILE_Pointer, "*************************************** :) YOUR INVOICE (: ***************************************

        fprintf(fILE_Pointer, "                          Customer Name    :  %s\n\n", user1.User_Name);
        fprintf(fILE_Pointer, "                          Customer Address :  %s\n\n", user1.User_address);
        fprintf(fILE_Pointer, "                          Contact  Details :  %lu\n\n", user1.User_contactdetails);
        fprintf(fILE_Pointer, "                          Email-id         :  %s\n\n", user1.User_emial_id);
        fprintf(fILE_Pointer, "                          Item name                Quantity       \n\n");
        for (int i = 0; i < count; i++)
        {
            fprintf(fILE_Pointer, "                          %s ----------> %d-only\n\n ", menu_array[item_array[i]], quantity_array[i
        }
        fprintf(fILE_Pointer, "--------------------------------------------------------------------------------------------------------------
        fprintf(fILE_Pointer, "                            Total             :   %3f Rs/- Only\n\n\n\n", grandTotal);
        /* Close file to save file data */
        fclose(fILE_Pointer);
    }

    void User ::Booking()
    {
        system("cls"); // this is the function which is defined in stlib.h and it is used to clear terminal screen
        std ::cout << "Here is the list of vacant and occupied tables\n\n";
        /*using the for loop to traverse through the array and check each value of element of array and print booked if it is 1 and
        for (int k = 0; k < 10; k++)
        {
            if (Table_array[k] == 1)
            {
                std ::cout << "Table no. " << k + 1 << " is already Booked\n";
            }
            else
            {
                std ::cout << "Table no. " << k + 1 << " is vacant\n";
            }
        }

        std ::cout << "Enter the table number of the table you want to book\n";
        std ::cin >> Table_no;
        Table_array[Table_no + 1] == 1; /*assigning 1 to [Table_no+1] of the array to mark it as booked */
        system("cls");                  // this is the function which is defined in stlib.h and it is used to clear terminal screen
        std ::cout << "                          ::::::::::::::::::::::::::::::::::::::::::::::::::::::\n";
        std ::cout << "                           Customer Name    :  " << user1.User_Name << "\n\n";
        std ::cout << "                           Contact  Details :  " << user1.User_contactdetails << "\n\n";
        std ::cout << "                           Email-id         :  " << user1.User_emial_id << "\n\n";
        std ::cout << "                           Congratulations table number " << Table_no << " successfully booked\n";
        std ::cout << "                          ::::::::::::::::::::::::::::::::::::::::::::::::::::::\n";
        save_booking_details(); // calling of function ----> save_booking_details()
        exit(0);                // exit(0) this function is used to successfully terminate and exit from the program
    }
    void User ::save_booking_details()
    {
        FILE *FILE_Pointer;                         // creation of file pointer
        FILE_Pointer = fopen("booking.txt", "a+"); // opening the file booking.txt in append plus mode and assigning it to file poi
        /* Write data to file */
        fprintf(FILE_Pointer, "                          ::::::::::::::::::::::::::::::::::::::::::::::::::::::\n");
        fprintf(FILE_Pointer, "                           Customer Name    :  %s\n\n", user1.User_Name);
        fprintf(FILE_Pointer, "                           Contact  Details :  %lu\n\n", user1.User_contactdetails);
        fprintf(FILE_Pointer, "                           Email-id         :  %s\n\n", user1.User_emial_id);
        fprintf(FILE_Pointer, "                           Congratulations table number %d successfully booked\n", Table_no);
        fprintf(FILE_Pointer, "                          ::::::::::::::::::::::::::::::::::::::::::::::::::::::\n");
        /* Close file to save file data */
        fclose(FILE_Pointer);
    }

    void User ::Cancel_Booking()
    {
        int cancel;
        system("cls"); // this is the function which is defined in stlib.h and it is used to clear terminal screen
        std ::cout << "Enter table number to cancel your booking \n\n";
        std ::cin >> cancel;
        Table_array[cancel + 1] == 0; /*assigning 0 to [cancel+1] of the array to mark it as vacant */
        system("cls");                // this is the function which is defined in stlib.h and it is used to clear terminal screen
        std ::cout << "                          ::::::::::::::::::::::::::::::::::::::::::::::::::::::\n\n";
        std ::cout << "                           Customer Name    :  " << user1.User_Name << "\n\n";

        std ::cout << "                           Contact  Details :  " << user1.User_contactdetails << "\n\n";
        std ::cout << "                           Email-id         :  " << user1.User_emial_id << "\n\n";
        std ::cout << "                           Your booking for table number " << cancel << " successfully canceled\n\n";
        std ::cout << "                          ::::::::::::::::::::::::::::::::::::::::::::::::::::::\n";
        exit(0); // exit(0) this function is used to successfully terminate and exit from the program
    }
    void User ::print_invoice()
    {
        FILE *file_pointer;                         // creation of file pointer
        file_pointer = fopen("invoice.txt", "r+"); // opening the file invoice.txt in read plus mode and assigning it to file point
        char c;                                     // declaration of char c to read data from the file
        c = fgetc(file_pointer);                    // fgetc is used to fetch data from the file
        /*while loop is used it will print data until the eof occurs*/
        while (c != EOF)
        {
            std ::cout << c;
            c = fgetc(file_pointer);
        }
        /* Close file to save file data */
        fclose(file_pointer);
        exit(0); // exit(0) this function is used to successfully terminate and exit from the program
    }
    void User ::view_bookings()
    {
        FILE *file1_pointer;                         // creation of file pointer
        file1_pointer = fopen("booking.txt", "r+"); // opening the file booking.txt in read plus mode and assigning it to file poin
        char c;                                    // declaration of char c to read data from the file
        c = fgetc(file1_pointer);                  // fgetc is used to fetch data from the file
        /*while loop is used it will print data until the eof occurs*/
        while (c != EOF)
        {
            std::cout << c;
            c = fgetc(file1_pointer);
        }
        fclose(file1_pointer); /* Close file to save file data */
        exit(0);               // exit(0) this function is used to successfully terminate and exit from the program
    }
```

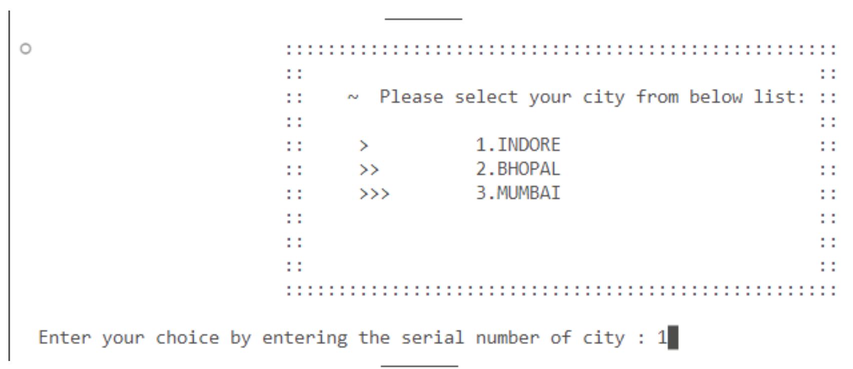
8 Code Output



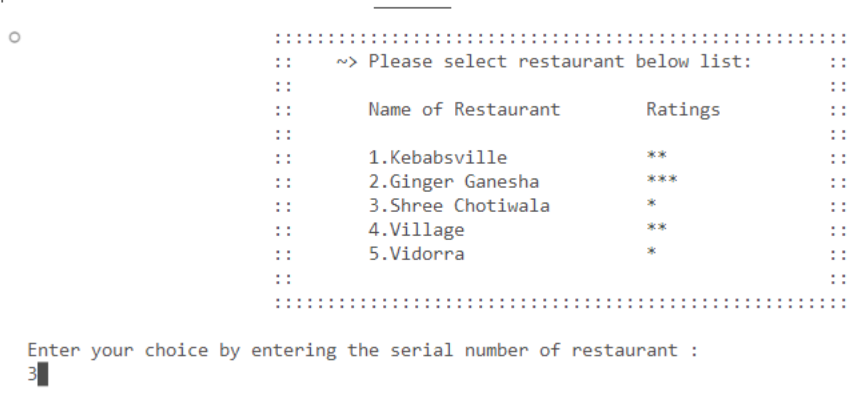
```
                    ----------------------------------------------------------------------------------------------------------------------

                                                              ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
                                                             :: THIS MINI PROJECT AIMS TO PERFORM VARIOUS FUNCTIONS ::
                                                              ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

                                                                         A.USER FUNCTIONS :
                                                                            -> ORDER FOOD :
                                                                            -> BOOK TABLE AT RESTAURANT :
                                                                            -> CANCELL BOOKING :
                                                                            -> GET INFORMATION ABOUT RESTAURANT :

                                                                         B.ADMIN FUNCTIONS :
                                                                            -> ADMIN AUTHENTICATION :
                                                                            -> SET/UPDATE GST RATES :
                                                                            -> GENERATE INVOICE :
                                                                            -> VIEW ALL INVOICES :
                                                                            -> VIEW ALL BOOKINGS :

Login:
For User login press 1:
For Admin login press 2:
To exit press 0:
1
```



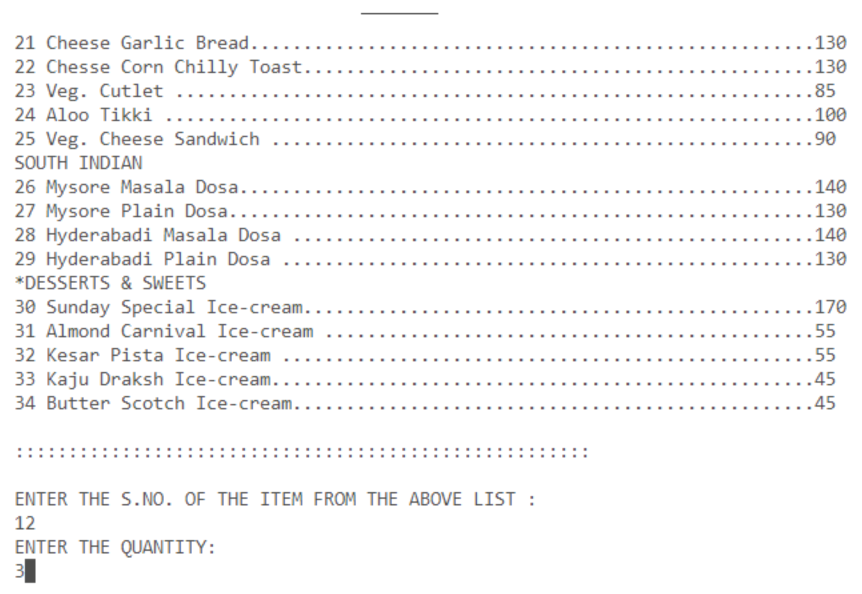
```
                        ::::::::::::::::::::::::::::::::::::::::::::::::::::::::
                        ::         !!!!!!!!!!!welcome!!!!!!!!!!!             ::
                        ::                                                   ::
                        ::    ~> Please select from below options:           ::
                        ::                                                   ::
                        ::       1.Get Information About Restaurant :>       ::
                        ::       2.Order Food:>                              ::
                        ::       3.Book table:>                              ::
                        ::       4.Cancel your Booking:>                     ::
                        ::       5.Exit:>                                    ::
                        ::::::::::::::::::::::::::::::::::::::::::::::::::::::::

   PLEASE SELECT FROM THE ABOVE OPTIONS:
```



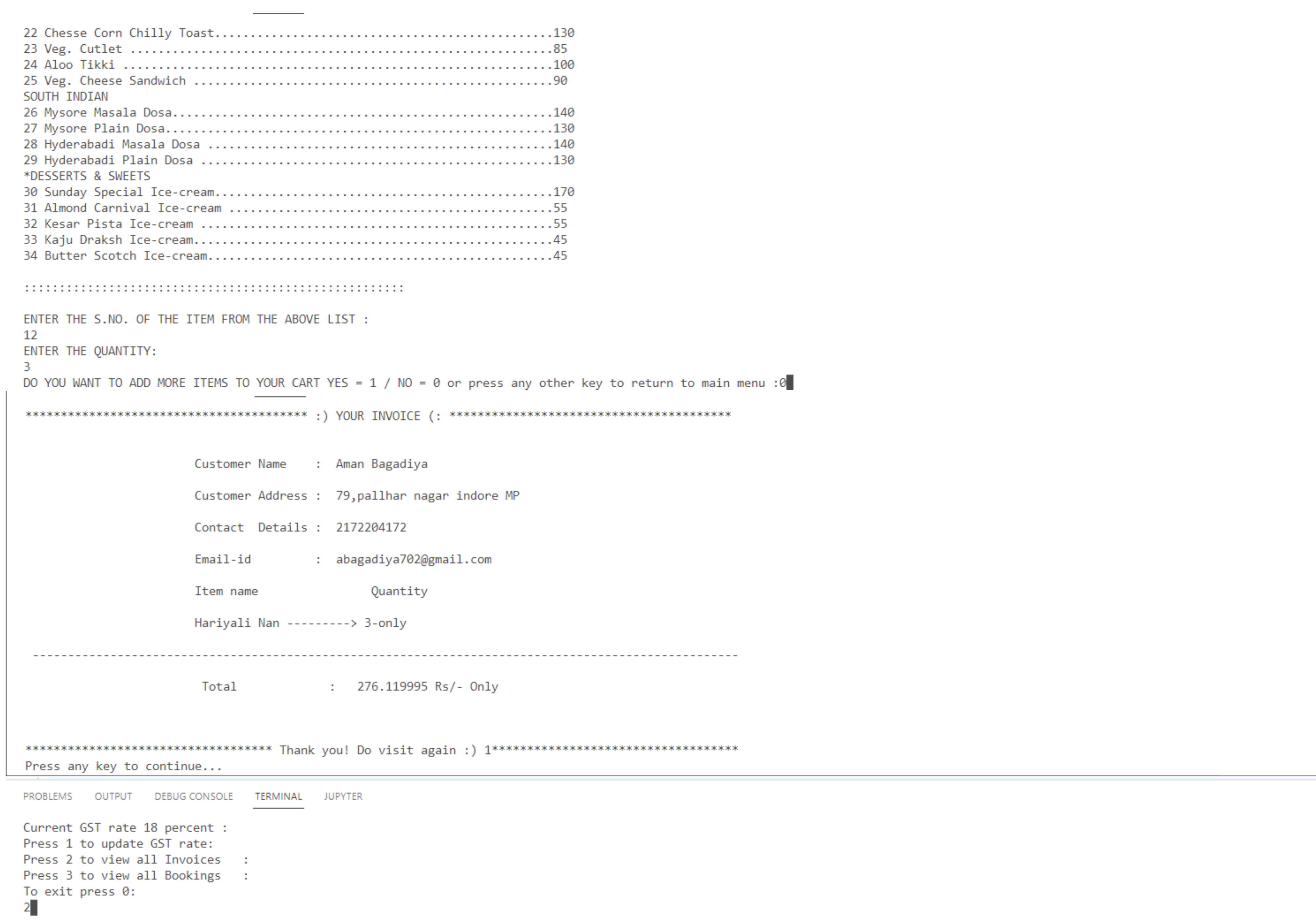
```
                       ::::::::::::::::::::::::::::::::::::::::::::::::::::
                       ::                                                ::
                       ::    ~  Please select your city from below list: ::
                       ::                                                ::
                       ::      >          1.INDORE                       ::
                       ::      >>         2.BHOPAL                       ::
                       ::      >>>        3.MUMBAI                       ::
                       ::                                                ::
                       ::                                                ::
                       ::                                                ::
                       ::::::::::::::::::::::::::::::::::::::::::::::::::::

 Enter your choice by entering the serial number of city : 1
```



```
                       ::::::::::::::::::::::::::::::::::::::::::::::::::::::
                       ::    ~> Please select restaurant below list:       ::
                       ::                                                  ::
                       ::       Name of Restaurant        Ratings          ::
                       ::                                                  ::
                       ::       1.Kebabsville             **               ::
                       ::       2.Ginger Ganesha          ***              ::
                       ::       3.Shree Chotiwala         *                ::
                       ::       4.Village                 **               ::
                       ::       5.Vidorra                 *                ::
                       ::                                                  ::
                       ::::::::::::::::::::::::::::::::::::::::::::::::::::::

 Enter your choice by entering the serial number of restaurant :
 3
```



```
21 Cheese Garlic Bread...................................................130
22 Chesse Corn Chilly Toast..............................................130
23 Veg. Cutlet ...........................................................85
24 Aloo Tikki ...........................................................100
25 Veg. Cheese Sandwich ..................................................90
SOUTH INDIAN
26 Mysore Masala Dosa....................................................140
27 Mysore Plain Dosa.....................................................130
28 Hyderabadi Masala Dosa ...............................................140
29 Hyderabadi Plain Dosa ................................................130
*DESSERTS & SWEETS
30 Sunday Special Ice-cream..............................................170
31 Almond Carnival Ice-cream .............................................55
32 Kesar Pista Ice-cream .................................................55
33 Kaju Draksh Ice-cream..................................................45
34 Butter Scotch Ice-cream................................................45

::::::::::::::::::::::::::::::::::::::::::::::::::::::

ENTER THE S.NO. OF THE ITEM FROM THE ABOVE LIST :
12
ENTER THE QUANTITY:
3
```



```
22 Chesse Corn Chilly Toast...............................................130
23 Veg. Cutlet ...........................................................85
24 Aloo Tikki ............................................................100
25 Veg. Cheese Sandwich ..................................................90
SOUTH INDIAN
26 Mysore Masala Dosa.....................................................140
27 Mysore Plain Dosa......................................................130
28 Hyderabadi Masala Dosa ................................................140
29 Hyderabadi Plain Dosa .................................................130
*DESSERTS & SWEETS
30 Sunday Special Ice-cream...............................................170
31 Almond Carnival Ice-cream .............................................55
32 Kesar Pista Ice-cream .................................................55
33 Kaju Draksh Ice-cream..................................................45
34 Butter Scotch Ice-cream................................................45

::::::::::::::::::::::::::::::::::::::::::::::::::::::

ENTER THE S.NO. OF THE ITEM FROM THE ABOVE LIST :
12
ENTER THE QUANTITY:
3
DO YOU WANT TO ADD MORE ITEMS TO YOUR CART YES = 1 / NO = 0 or press any other key to return to main menu :0

**************************************** :) YOUR INVOICE (: ****************************************


                        Customer Name    :  Aman Bagadiya

                        Customer Address :  79,pallhar nagar indore MP

                        Contact  Details :  2172204172

                        Email-id         :  abagadiya702@gmail.com

                        Item name                Quantity

                        Hariyali Nan ---------> 3-only

 ----------------------------------------------------------------------------------------------------

                         Total             :   276.119995 Rs/- Only


*********************************** Thank you! Do visit again :) 1***********************************
Press any key to continue...
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
Current GST rate 18 percent :
Press 1 to update GST rate:
Press 2 to view all Invoices   :
Press 3 to view all Bookings   :
To exit press 0:
2
```