COMPUTER SCIENCE PROJECT

CLASS XII A (2017-18)

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT OF C.B.S.E., DELHI

TOPIC- " A Hacker's System Database "

SUBMITTED BY:	UNDER THE GUIDANCE OF:
NAME – SAKET PANDEY	Mr. Harsh Sinha
ROLL NO - __________	

DELHI PUBLIC SCHOOL RAIPUR

Delhi Public School Raipur (CG), Semariya Baloda Bazaar Road Raipur-492001 (C.G.)

Ph.: +91-771-2284804. Email: dpsraipur@gmail.com. Website: www.dps.raipur.com

CERTIFICATE

This is to certify that SAKET PANDEY, a student of Class XII, Delhi Public School Raipur, has successfully completed the project of Computer Science as prescribed by CBSE in the year 2017-18.

This project is absolutely genuine and does not indulge in plagiarism of any kind.

Date:

Roll No:

Signature of Internal Examiner *Signature of External Examiner*

ACKNOWLEDGEMENT

I feel proud to present my project in Computer Science on the topic “**A Hacker’s System Database** “.

I would like to express my deep sense of thanks and gratitude to my project guide **Mr. Harsh Sinha Sir** for guiding me immensely through the course of the project. He always evinced keen interest in my work. His constructive advice and constant motivation have been responsible for the successful completion of this project.

Rigorous hard work has been put in this project to ensure that it proves to be the best.

Last but not the least; I would like to thank my parents for their motivation and support. I must also thank my classmates for their timely help and support for completion of this project.

SAKET PANDEY
Class - XII A

CONTENTS

HEADER FILES USED

a) **FSTREAM.H –** for file handling, cin and cout.

b) **CONIO.H –** for clrscr() and getch() functions.

c) **STDIO.H –** for standard I/O operations.

d) **PROCESS.H –** for exit() function.

e) **STRING.H –** for string handling.

f) **IOMANIP.H –** for setw() function.

g) **DOS.H –** for delay() function.

WORKING DESCRIPTION

This program is designed such that it represents a replica of a Hackers Database.
This program consists of the following options –

1. **TO ADD CUSTOMER DATA FOR CONTRACT PURPOSES**
2. **TO MODIFY, VIEW OR DELETE ABOVE DATA**
3. **TO REQUEST A CONSIGNMENT ONLY BY A REGISTERED CUSTOMER**
4. **TO MODIFY VIEW OR DELETE ABOVE DATA**
5. **TO INPUT CONSIGNMENT STATUS**
6. **TO VIEW THE REGIONS OF OPERATION**
7. **TO VIEW AND UPDATE PERSONAL INVENTORY**

✚ CODING

NOTE :

[Username – “**saket**”

Or “**mihir**”

Or “**archisman**”

Password – “**unknown**”]

```
     //A HACKERS SYSTEM DATABASE

#include<fstream.h>
#include<conio.h>
#include<stdio.h>
#include<process.h>
#include<string.h>
#include<iomanip.h>
#include<dos.h>

//____________________________________________CONTRACTS_______________________________


//*****************************************************************
//              CLASS - 1 USED FOR CONTRACTS
//******************************************************************


class consignment
{
char cno[10];
char task[70];
int dline;
char mode[20];
char amount[80];
  public:
                                        void add_consignment()
                                        {
                                            cout<<"\n\n\t Enter a Consignment No. : ";
                                            cin>>cno;
                                            cout<<"\n\t Specify Task [Max. Char=25] -  ";
                                            gets(task);
                                            cout<<"\n\t Enter the Deadline for Task Completion(in
days) :  ";
                                            cin>>dline;
                                            cout<<"\n\t Enter the Mode Of Payment \n(Cash/Credit
Card/Bitcoin/PayPal/Wire Transfer):  ";
                                            gets(mode);
                                            cout<<"\n\n\t Amount(in $):  ";
                                            gets(amount);
                                            cout<<"\n\n\n\t CONSIGNMENT CREATED!";
                                            getch();
```

```
                }

                void show_cons()
                {
                    cout<<"\n\t Consignment  No. : "<<cno;
                    cout<<"\n\t TASK - ";
                    puts(task);
                    cout<<"\n\t Deadline : "<<dline;
                    cout<<"\n\t Payment Mode : ";
                    puts(mode);
                    cout<<"\n\t Amount(in $): ";
                    puts(amount);
                    getch();
                }

                void update_cons()
                {
                    cout<<"\n\t Consignment  No. : "<<cno;
                    cout<<"\n\t Update TASK - ";
                    gets(task);
                    cout<<"\n\t Update Deadline : ";
                    cin>>dline;
                    cout<<"\n\t Update Amount : ";
                    gets(amount);
                    getch();
                }

                char* retcno()
                {
                    return cno;
                }

                int retdline()
                {
                    return dline;
                }

                void report()
                {cout<<cno<<"\t"<<setw(10)<<task<<setw(28)<<mode<<setw
(14)<<amount<<endl;}


};      //class ends here
```

```
class customer
{
                        char custid[6];
                        char name[20];
                        char ctcno[6];
                        int token;
public:
                        void create_customer()
                        {
                            clrscr();
                            cout<<"\n\t NEW CUSTOMER ENTRY...\n";
                            cout<<"\n\t Enter The Customer ID:  ";
                            cin>>custid;
                            cout<<"\n\n\t Customer Name:  ";
                            gets(name);
                            token=0;
                            ctcno[0]='/0';
                            cout<<"\n\n\t Customer Record Created..";
                            getch();
                        }

                        void show_customer()
                        {
                            cout<<"\n\t Customer ID:  "<<custid;
                            cout<<"\n\t Name:  ";
                            puts(name);
                            cout<<"\n\t No of Consignments: "<<token;
                            if(token==1)
                                    cout<<"\n\t Consignment No "<<ctcno;
                        }

                        void modify_customer()
                        {
                            cout<<"\n\t Customer ID: "<<custid;
                            cout<<"\n\t Modify Customer Name : ";
                            gets(name);
                        }

                        char* retcustid()
                        {
                            return custid;
                        }
```

```
                                        char* retctcno()
                                        {
                                            return ctcno;
                                        }

                                        int rettoken()
                                        {
                                            return token;
                                        }

                                        void addtoken()
                                        {token=1;}

                                        void resettoken()
                                        {token=0;}

                                        void getctcno(char t[])
                                        {
                                            strcpy(ctcno,t);
                                        }

                                        void report()
                                        {cout<<"\t"<<custid<<setw(25)<<name<<setw(30)<<token<<e
ndl;}

};      //class ends here


//**************************************************************
//         global declaration for stream object, object
//***************************************************************

fstream fp,fp1,fp10;
consignment cns;
customer ct;


//**************************************************************
//               function to write in file
//***************************************************************

void write_consignment()
{
                                         write:;
                                        char ch;
char sn[6];
int found=0;
clrscr();
cout<<"\n\nNEW CONSIGNMENT ...";
cout<<"\n\n\tEnter The Customer ID : ";
cin>>sn;
fp.open("customer.dat",ios::in|ios::out|ios::binary);
fp1.open("consignment.dat",ios::in|ios::out|ios::app|ios::binary);
while(fp.read((char*)&ct,sizeof(customer)) && found==0)
{
    if(strcmpi(ct.retcustid(),sn)==0)
    {
        found=1;
        if(ct.rettoken()==0)
        {
            cns.add_consignment();
            fp1.write((char*)&cns,sizeof(consignment));
            fp1.read((char*)&cns,sizeof(consignment));
            {
                    cns.show_cons();
                    ct.addtoken();
                    ct.getctcno(cns.retcno());
                    int pos=-1*sizeof(ct);
                    fp.seekp(pos,ios::cur);

fp.write((char*)&ct,sizeof(customer));
                    cout<<"\n\n\t Consignment Request Successful.\n\n";
                    }

        }
        else
            cout<<"\t Last Consignment Pending! ";

    }
}
if(found==0)
    cout<<"\n\n Customer does not exist in System Database. First CREATE Customer Record!";
getch();
cout<<"\n\nDo you want to add more records..(Y/N?)";
    cin>>ch;
    if(ch=='y'||ch=='Y')
     { goto write;
```

```
                                         }
                                    fp.close();
                                    fp1.close();
}

void write_customer()
{
                                    char ch;
                                    fp.open("customer.dat",ios::out|ios::app|ios::binary);
                                    write2:;
                                        ct.create_customer();
                                        fp.write((char*)&ct,sizeof(customer));
                                        cout<<"\n\nDo you want to add more records..(Y/N)?";
                                        cin>>ch;
                                        if(ch=='y'||ch=='Y')
                                         { goto write2;
                                         }
                                    fp.close();
}


//************************************************************
//          function to read specific record from file
//*************************************************************

void display_sp_cons(char n[])
{
                                    cout<<"\n\t CONSIGNMENT DETAILS\n";
                                    int flag=0;
                                    fp1.open("consignment.dat",ios::in|ios::binary);
                                    while(fp.read((char*)&cns,sizeof(consignment)))
                                    {
                                        if(strcmpi(cns.retcno(),n)==0)
                                        {
                                               cns.show_cons();
                                               flag=1;
                                        }
                                    }

                                    fp1.close();
                                    if(flag==0)
                                        cout<<"\n\n\t No such Consignment exists";
                                    getch();
}
```

```
void display_sp_cust(char n[])
{
                                        cout<<"\n\t CUSTOMER DETAILS\n";
                                        int flag=0;
                                        fp.open("customer.dat",ios::in|ios::binary);
                                        while(fp.read((char*)&ct,sizeof(customer)))
                                        {
                                            if((strcmpi(ct.retcustid(),n)==0))
                                            {
                                                    ct.show_customer ();
                                                    flag=1;
                                            }
                                        }

                                        fp.close();
                                        if(flag==0)
                                            cout<<"\n\n\t Customer does not exist";
                                        getch();
}


//****************************************************************
//              function to modify record of file
//*****************************************************************

void update_cons()
{
                                        char n[6];
                                        int found=0;
                                        clrscr();
                                        cout<<"\n\n\tMODIFY CONSIGNMENT RECORD.... ";
                                        cout<<"\n\n\tEnter The Consignment No. :";
                                        cin>>n;
                                        fp1.open("consignment.dat",ios::in|ios::out|ios::binary);
                                        while(fp1.read((char*)&cns,sizeof(consignment)) &&
found==0)
                                        {
                                            if(strcmpi(cns.retcno(),n)==0)
                                            {
                                                    cns.show_cons();
                                                    cout<<"\n\t Enter The New Details of
Consignment"<<endl;
                                                    cns.update_cons();
                                                    int pos=-1*sizeof(consignment);
                                                    fp1.seekp(pos,ios::cur);
```

```
                                    fp1.write((char*)&cns,sizeof(consignment));
                                    cout<<"\n\n\t Record Updated";
                                    found=1;
                               }
                           }

                           fp1.close();
                           if(found==0)
                               cout<<"\n\n\t Record Not Found ";
                           getch();
}


void modify_customer()
{
                           char n[6];
                           int found=0;
                           clrscr();
                           cout<<"\n\n\tMODIFY CUSTOMERRECORD... ";
                           cout<<"\n\n\tEnter The Customer ID: ";
                           cin>>n;
                           fp.open("customer.dat",ios::in|ios::out|ios::binary);
                           while(fp.read((char*)&ct,sizeof(customer)) && found==0)
                           {
                               if(strcmpi(ct.retcustid(),n)==0)
                               {
                                    ct.show_customer();
                                    cout<<"\n\t Enter the New Details of
Customer"<<endl;
                                    ct.modify_customer();
                                    int pos=-1*sizeof(ct);
                                    fp.seekp(pos,ios::cur);
                                    fp.write((char*)&ct,sizeof(customer));
                                    cout<<"\n\n\t Record Updated";
                                    found=1;
                               }
                           }

                           fp.close();
                           if(found==0)
                               cout<<"\n\n\t Record Not Found ";
                           getch();
}
```

```
//************************************************************
//                    function to delete record of file
//*************************************************************

void delete_customer()
{
                                    char n[6];
                                    int flag=0;
                                    clrscr();
                                    cout<<"\n\n\n\tDELETE CUSTOMER ..";
                                    cout<<"\n\n\t Enter The Customer ID of the Customer You
Want To Delete : ";
                                    cin>>n;
                                    fp.open("customer.dat",ios::in|ios::out|ios::binary);
                                    fstream fp2;
                                    fp2.open("Temp.dat",ios::out|ios::binary);
                                    fp.seekg(0,ios::beg);
                                    while(fp.read((char*)&ct,sizeof(customer)))
                                    {
                                        if(strcmpi(ct.retcustid(),n)!=0)
                                                fp2.write((char*)&ct,sizeof(customer));
                                        else
                                                flag=1;
                                    }

                                    fp2.close();
                                    fp.close();
                                    remove("customer.dat");
                                    rename("Temp.dat","customer.dat");
                                    if(flag==1)
                                        cout<<"\n\n\tRecord Deleted ..";
                                    else
                                        cout<<"\n\n\t Record not found!!";
                                    getch();
}


void remove_cons()
{
                                    char n[6];
                                    clrscr();
                                    cout<<"\n\n\n\tDELETE CONSIGNMENT ...";
                                    cout<<"\n\n\t Enter The Consignment No. you Want To Delete
: ";
```

```
                        cin>>n;
                        fp1.open("consignment.dat",ios::in|ios::out|ios::binary);
                        fstream fp2;
                        fp2.open("Temp.dat",ios::out|ios::binary);
                        fp1.seekg(0,ios::beg);
                        while(fp.read((char*)&cns,sizeof(consignment)))
                        {
                            if(strcmpi(cns.retcno(),n)!=0)
                            {
                                    fp2.write((char*)&cns,sizeof(consignment));
                            }
                        }

                        fp2.close();
                        fp1.close();
                        remove("consignment.dat");
                        rename("Temp.dat","consignment.dat");
                        cout<<"\n\n\tRecord Deleted ..";
                        getch();
}



//************************************************************
//                  DISPLAY ALL CUSTOMERS LIST
//*************************************************************

void display_all_cust()
{
                        clrscr();
                        fp.open("customer.dat",ios::in|ios::binary);
                        if(!fp)
                        {
                            cout<<"\t\t\n ERROR!!! FILE COULD NOT BE OPEN ";
                            getch();
                            return;
                        }

                        cout<<"\n\n\t\t\tCUSTOMER LIST\n\n";
                        cout<<"---------------------------------------------------------------------------
\n";
                        cout<<"\tCustomer
ID"<<setw(15)<<"NAME"<<setw(30)<<"Consignments Pending\n";
```

```
                                    cout<<"________________________________________________________
__________________________\n";

                                    while(fp.read((char*)&ct,sizeof(customer)))
                                    {
                                        ct.report();
                                    }

                                    fp.close();
                                    getch();
}



//****************************************************************
//          Function to display consignments list
//*****************************************************************

void display_all_cons()
{
                                    clrscr();
                                    fp1.open("consignment.dat",ios::in|ios::binary);
                                    if(!fp1)
                                    {
                                        cout<<"ERROR!!! FILE COULD NOT BE OPEN ";
                                        getch();
                                        return;
                                    }

                                    cout<<"\n\n\t\t\tCONSIGNMENT LIST\n\n";
                                    cout<<"=============================================
=====================================\n";
                                    cout<<"Cons
No."<<setw(8)<<"TASK"<<setw(37)<<"PaymentMode"<<setw(16)<<"Amount($))\n";
                                    cout<<"=============================================
=====================================\n";

                                    while(fp1.read((char*)&cns,sizeof(consignment)))
                                    {
                                        cns.report();
                                    }
                                    fp1.close();
                                    getch();
}
```

```
//****************************************************************
//              Function to input TASK COMPLETED>
//*****************************************************************

void complete_cons()
{
  char sn[6],bn[6];
  int found=0,flag=0,day,loss;
  clrscr();
  cout<<"\n\n\tConsignment STATUS UPDATE...";
  cout<<"\n\n\tEnter Customer ID: ";
  cin>>sn;
  fp.open("customer.dat",ios::in|ios::out|ios::binary);
  fp1.open("consignment.dat",ios::in|ios::out|ios::binary);
  while(fp.read((char*)&ct,sizeof(customer)) && found==0)
    {
                                   if(strcmpi(ct.retcustid(),sn)==0)
                                   {
                                       found=1;
                                       if(ct.rettoken()==1)
                                       {

                                 while(fp1.read((char*)&cns,sizeof(consignment))&& flag==0)
                                            {
                                             if(strcmpi(cns.retcno(),ct.retctcno())==0)
                                            {
                                                  cns.show_cons();
                                                  flag=1;
                                                  cout<<"\n\n\t Consignment completed in no.
of days- ";

                                                  cin>>day;
                                                  if(day>(cns.retdline()))
                                                  {
                                                    loss=(day-(cns.retdline()))*1;
                                                    cout<<"\n\n\t Loss Incurred- "<<loss;
                                                  }
                                                         ct.resettoken();
                                                         int pos=-1*sizeof(ct);
                                                         fp.seekp(pos,ios::cur);

                                 fp.write((char*)&ct,sizeof(customer));
```

```
                                                            cout<<"\n\n\t Consignment
Completed Successfullly.";
                                                       }
                                                  }
                                                   }
                                                else
                                                       cout<<"\t NO Consignment exists under this ID.";
                                             }
                                           }
   if(found==0)
                                   cout<<"\t Invalid Customer ID!";
                                   getch();
 fp.close();
 fp1.close();
 }

//************************************************************
//           CONNTRACTS MENU FUNCTION
//*************************************************************



void condet()
{
                                   clrscr();
                                    con:;
                                   int ch2;
                                   cout<<"\n\n\n\t________CONTRACTS______ ";
                                   cout<<"\n ----------------------------------------------";
                                   cout<<"\n\n\t1.CREATE CUSTOMER RECORD";
                                   cout<<"\n\n\t2.Display All Customer Records";
                                   cout<<"\n\t3.Display Specific Customer Record ";
                                   cout<<"\n\n\t4.Modify Customer Details";
                                   cout<<"\n\t5.Delete Customer Record";
                                   cout<<"\n -----------------------------------------------";
                                   cout<<"\n\n\t6.ADD NEW CONSIGNMENT ";
                                   cout<<"\n\n\t7.DISPLAY All Consignments ";
                                   cout<<"\n\t8.Display Specific Consignment Record";
                                   cout<<"\n\n\t9.Update Consignment Record ";
                                   cout<<"\n\t10.Consignment Completion Status Update ";
                                   cout<<"\n\t11.Remove Consignment";
                                   cout<<"\n -----------------------------------------------";
                                   cout<<"\n\n\t12.EXIT";
                                   cout<<"\n\n\tPlease Enter Your Choice (1-12) - ";
                                   cin>>ch2;
```

```
switch(ch2)
{
    case 1: clrscr();
           write_customer();
           break;
    case 2:  display_all_cust();
           break;
    case 3:{
           char num[6];
           clrscr();
           cout<<"\n\n\tPlease Enter The Customer ID. ";
           cin>>num;
           display_sp_cust(num);
           break;
           }
    case 4: modify_customer();
           break;
    case 5: delete_customer();
           break;
    case 6: clrscr();
           write_consignment();
           break;
    case 7: display_all_cons();
           break;
    case 8: {
           char num[6];
           clrscr();
           cout<<"\n\n\tPlease Enter The Consignment No. ";
           cin>>num;
           display_sp_cons(num);
           break;
           }
    case 9: update_cons();
           break;
    case 10:complete_cons();
           break;
    case 11: remove_cons();
           break;
    case 12: return;
    default:
                  { cout<<"\n\t\t Wrong choice Entered!! Try
Again..";
           getch();
           goto con; }
}
```

```
                                        condet();
}




//__________________________REGIONS OF OPERATION___________________________



//***************************************************************
//       FUNCTION FOR - REGIONS OF OPERATION
//****************************************************************


 void regions_of_operation()
{
                                        clrscr();
                                        cout<<"\n\t\t\tRegions Of Operation\n\n\n";

                                        cout<<"===========================================
===================================";
                                            cout<<"\nCOUNTRIES"        <<setw(66)<<"TYPE (HQ =
HeadQuarters)";

                                        cout<<"\n==========================================
====================================";
                                            cout<<"\n\n RUSSIA"          <<setw(60)<<"HQ+Server 1";
                                            cout<<"\n\n Brazil"            <<setw(58)<<"Server 2";
                                            cout<<"\n\n Chile"          <<setw(64)<<"Operator
Available";
                                            cout<<"\n\n India"              <<setw(59)<<"Server 3";
                                            cout<<"\n\n IRAQ"          <<setw(62)<<"HQ+Server 4";
                                            cout<<"\n\n Ireland"      <<setw(62)<<"Operator
Available";
                                            cout<<"\n\n North Korea"  <<setw(58)<<"Operator
Available";
                                            cout<<"\n\n SPAIN"         <<setw(56)<<"HQ";
                                            cout<<"\n\n United States"        <<setw(54)<<"Hardware
Cache";
                                            cout<<"\n\n United Arab Emirates"
<<setw(49)<<"Operator Available";
```

```
                                    cout<<"\n\n*******************************************************
************************";
                                    getch();
}



//____________________________________INVENTORY_______________________________



//***************************************************************
//          CLASS-2 BEGINS FOR INVENTORY
//****************************************************************

class inventory
{
                                    char item[50];
                                    char quan[20];
public:
                                    void create_inventory()
                                    {
                                        clrscr();
                                        cout<<"\n\t NEW INVENTORY ITEM...\n";
                                        cout<<"\n\t Enter Item:  ";
                                        gets(item);
                                        cout<<"\n\t Quantity:  ";
                                        gets(quan);
                                        cout<<"\t \n\nItem Added to Inventory!.";
                                        getch();
                                    }

                                    void show_inventory()
                                    {
                                        cout<<"\n\t Item:  ";
                                        puts(item);
                                        cout<<"\n\t Quantity: "<<quan;
                                    }


                                    char* retitem()
                                    {
                                        return item;
                                        }

                                        void report()
                                        {cout<<"\n"<<item<<setw(53)<<quan<<endl;}

};      //class ends here



//***************************************************************
//      Global declaration for stream object, object
//****************************************************************

inventory inv;

//***************************************************************
//                      Add to Inventory
//****************************************************************

void write_inventory()
{
                                        char ch;
                                        fp10.open("inventory.dat",ios::out|ios::app|ios::binary);
                                        write10:;
                                            inv.create_inventory();
                                            fp10.write((char*)&inv,sizeof(inventory));
                                            cout<<"\n\nDo you want to add more records..(Y/N)?";
                                            cin>>ch;
                                            if(ch=='y'||ch=='Y')
                                             { goto write10;
                                             }
                                        fp10.close();
}

//***************************************************************
//                  Display All from Inventory
//****************************************************************

void display_inventory()
{
                                        clrscr();
                                        fp10.open("inventory.dat",ios::in|ios::binary);

                                        cout<<"\n\n\t\t\tITEM LIST\n\n";
```

```
                         cout<<"----------------------------------------------------------------
\n";
                         cout<<"\tITEMS"<<setw(50)<<"QUANTITY\n";
                         cout<<"________________________________________
__________________\n";
                         cout<<"\n\n ASUS RoG - Core i9 3.2GHz/16 GBDDR4/128
SSD"        <<setw(18)<<12;
                         cout<<"\n\n HardDrive 2 TB"         <<setw(47)<<62;
                         cout<<"\n\n JBL Audio System J0021"
<<setw(39)<<"02";
                         cout<<"\n\n Lamborghini Huracan 720HP"

<<setw(37)<<"01\n";
                         while(fp10.read((char*)&inv,sizeof(inventory)))
                         {
                            inv.report();
                         }

                         fp10.close();
                         getch();
}

//***************************************************************
//              Delete from Inventory
//***************************************************************

void delete_inventory()
{
                         char n[30];
                         int flag=0;
                         clrscr();
                         cout<<"\n\n\n\tDELETE ITEM FROM Inventory..";
                         cout<<"\n\nEnter The Item You Want To Delete : ";
                         cin>>n;
                         fp10.open("inventory.dat",ios::in|ios::out|ios::binary);
                         fstream fp20;
                         fp20.open("Temp2.dat",ios::out|ios::binary);
                         fp10.seekg(0,ios::beg);
                         while(fp.read((char*)&inv,sizeof(inventory)))
                         {
                            if(strcmpi(inv.retitem(),n)!=0)
                                 fp20.write((char*)&inv,sizeof(inventory));
                            else
                                 flag=1;
                         }
```

```
                                    fp20.close();
                                    fp10.close();
                                    remove("inventory.dat");
                                    rename("Temp2.dat","inventory.dat");
                                    if(flag==1)
                                        cout<<"\n\n\tItem Deleted ..";
                                    else
                                        cout<<"\n\nItem not found!!";
                                    getch();
}

//**************************************************************
//             INVENTORY MENU FUNCTION
//***************************************************************

void invdet()
{
                                    clrscr();
                                     inv:;
                                    int ch2;
                                    cout<<"\n\n\n\t\t\t________INVENTORY MENU______ ";
                                    cout<<"\n\t\t ------------------------------------------------";
                                    cout<<"\n\n\t\t\t1.VIEW Inventory";
                                    cout<<"\n\n\t\t\t2.Add Item to Inventory";
                                    cout<<"\n\t\t\t3.Delete Item from Inventory ";
                                    cout<<"\n\n\t\t\t4.EXIT to Main Menu";
                                    cout<<"\n\t\t ______________________________________";
                                    cout<<"\n\n\t\t\tEnter Your Choice (1-4) - ";
                                    cin>>ch2;
                                    switch(ch2)
                                    {
                                        case 1:  clrscr();
                                               display_inventory();
                                               break;
                                        case 2:  write_inventory();
                                               break;

                                        case 3: delete_inventory();
                                               break;
                                        case 4: return;
                                        default:
                                                      { cout<<"\n\t\t\t Wrong choice Entered!! Try
Again..";
                                               getch();
                                               goto inv; }
                                        }
                                        invdet();
}




//______________________________WELCOME SCREEN_______________________________



//************************************************************
//                 INTRO SCREEN Login Window
//*************************************************************

void intro()
{
clrscr();
textcolor(WHITE);
int k=0;
char str[20],user[20];
 clrscr();
                                        cout<<"\n\nYou have reached the Database of - ";
                                        cout<<"\n\n\n\t\t\t_________________________";
                                        cout<<"\n\n\t\t\t|\t ANONYMOUS\t  |";
                                        cout<<"\n\t\t\t_________________________";
                                        cout<<"\n\n\n\t We Are Legion. We Do not Forgive. We Do not
Forget. Expect Us!\n\n";
                                        cout<<"\t\t\t   Active Since :-  2004..\n\n";
                                        cout<<"\n\n\n\n\n\t Project by - SAKET PANDEY, ARCHISMAN
HOTA & MIHIR TIWARI.";
 getch();

 //USERNAME
  user:;
 clrscr();
                                        cout<<"\n\n\n\n\t\t\t username - ";
                                        cin>>user;
                                        if(strcmp(user,"saket")!=0 && strcmp(user,"mihir")!=0 &&
strcmp(user,”archisman")!=0)
                                         { cout<<"\n\tSorry! Permission denied.";
                                          getch();
```

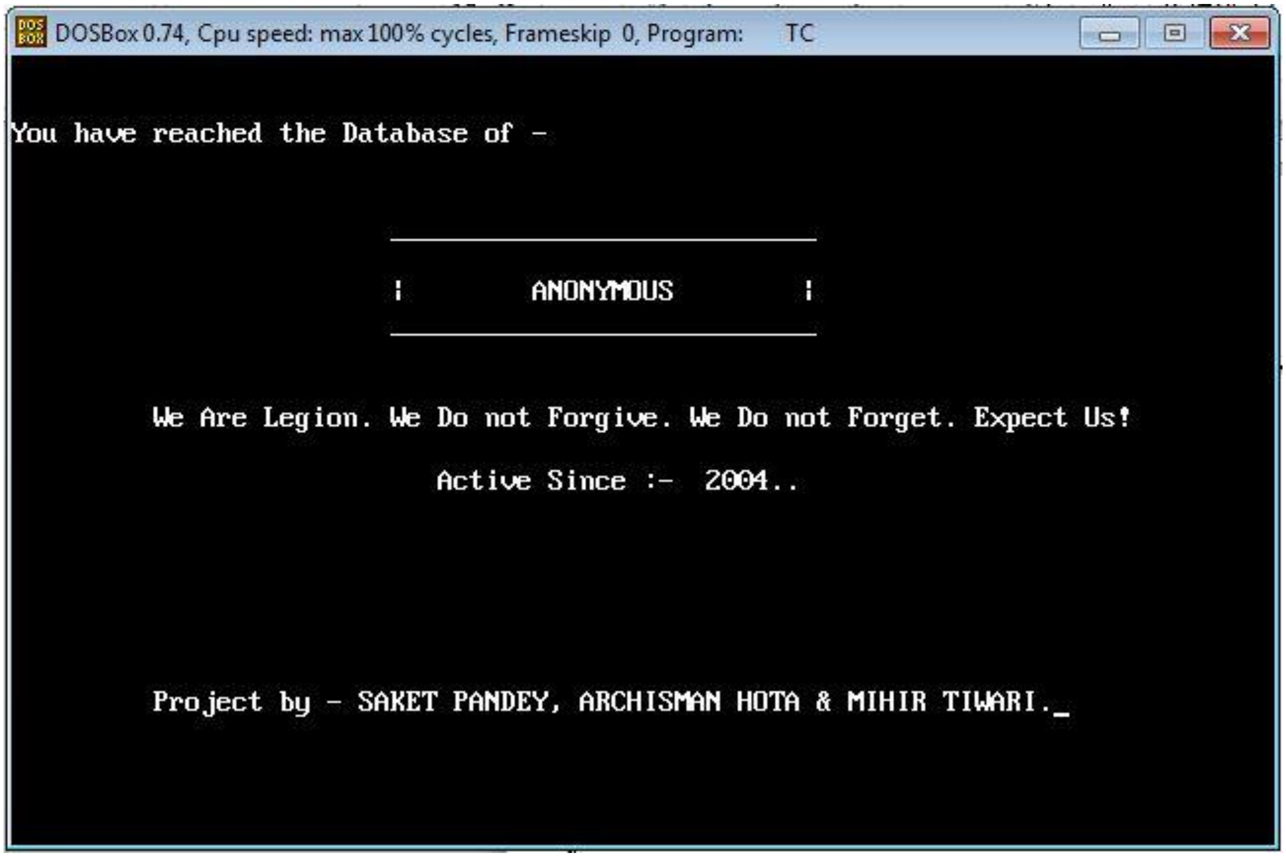
```
                                  goto user;
                                 }
 //PASSWORD
 pass:;
cout<<"\n\t\t\tpassword(7) - ";
str[0]=getch();
cout<<"*";
str[1]=getch();
cout<<"*";
str[2]=getch();
cout<<"*";
str[3]=getch();
cout<<"*";
str[4]=getch();
cout<<"*";
str[5]=getch();
cout<<"*";
str[6]=getch();
cout<<"*";
str[7]='\0';
cout<<"*";
if(strcmp(str,"unknown")==0)
 {
//INTRO LOOP
 clrscr();
                                 int n=101;
                                 for(int i=1;i<=n;i++)
                                 {
                                     cout<<"\n\t\t";
                                     for(int j=1;j<=(n-75);j++)
                                      cout<<"\t"<<n<<"01";
                                      cout<<"\t __";
                                 }
                                 cout<<"\n\n\n\n";
                                 delay(100);
                                 textcolor(GREEN);cprintf("                  --------^ACCESS
GRANTED^---------");

                                 textcolor(WHITE);
 delay(550);
 cout<<"\n\n\n\n\n\n\n\t\t ______________________<LOADING>";
 cout<<"_";
 delay(150);
 cout<<"_";
 delay(150);
 cout<<"_";
 delay(150);
 cout<<"_";
 delay(50);
 cout<<"_";
 delay(50);
 cout<<"_";
 delay(50);
 cout<<"_";
 delay(250);
 cout<<"_";
 delay(50);
 cout<<"_";
 delay(50);
 cout<<"_";
 delay(50);
 cout<<"_";
 delay(50);
 cout<<"_";
 delay(50);
 cout<<"_";
 delay(50);
 cout<<"_";
 delay(50);
 cout<<"__";
 delay(150);
 cout<<"__";
 delay(250);
 cout<<"__\n\n\n";
 delay(350);
 clrscr();

 }
else
 { cout<<"\n\n\t\t\t### Password INCORRECT. ###\n";      //Variable 'k' USED.
 k++;
 getch();

    if(k==2)
                                                { textcolor(RED+BLINK); cprintf("\n Exiting the Module!!
Bye..\n");
                                                  textcolor(WHITE);
                                                  getch();
                                                  exit(0);
                                                }
    else
```

```
                        {
                        cout<<"\n\n\t\t__________ Re-enter the password _________\n";
                        goto pass;
                        }
  }
}




//******************************************************************
//                THE MAIN FUNCTION OF PROGRAM
//*******************************************************************

void main()
{
                        textcolor(WHITE);
                        char ch;
                        intro();
                            choice:;
                           clrscr();
                           textcolor(LIGHTBLUE); cprintf("\r\n \r\n
^^^^^# ENCRYPTED DATABASE #^^^^^^\r\n");
                           textcolor(WHITE);
                           cout<<"  \t\t\t________________________________________";
                           cout<<"\n\n\n\n\t\t\t*     1.CONTRACTS              *";
                           cout<<"\n\n\n\t\t\t*    2.Regions Of Operation      *";
                           cout<<"\n\n\n\t\t\t*    3.Inventory              *";
                           cout<<"\n\n\n\t\t\t*    4.EXIT                 *\n";
                           cout<<"\n\t\t\t________________________________________";
                           cout<<"\n\n\n\t Enter your choice(1-4): ";
                           ch=getch();
                           switch(ch)
                           {
                                case '1':clrscr();
                                      condet();
                                      goto choice;
                                case '2': clrscr();
                                       regions_of_operation();
                                       goto choice;
                                case '3':clrscr();
                                      invdet();
                                      goto choice;
                                case '4': goto end;
```

```
                                                        default :
                                                                textcolor(RED+BLINK); cprintf(" Wrong
Choice Entered!! Try Again.");
                                                                textcolor(WHITE);
                                                                getch();
                                                                goto choice;
                                                }
                                                 end:;
                                                clrscr();
                                                textcolor(LIGHTBLUE); cprintf("\r\n \r\n \r\n \r\n \r\n \r\n-----
------------------------------------------------------------------");
                                                cout<<"\n\t\t\t THANK YOU FOR USING THIS PROJECT! ---
- ";
                                                textcolor(LIGHTRED); cprintf("\r\n\r\n--------------------------
--------------------------------------------");
                                                textcolor(WHITE);
                                                getch();
}



//________________________________PROGRAM ENDS________________________________
```

OUTPUT SCREENS

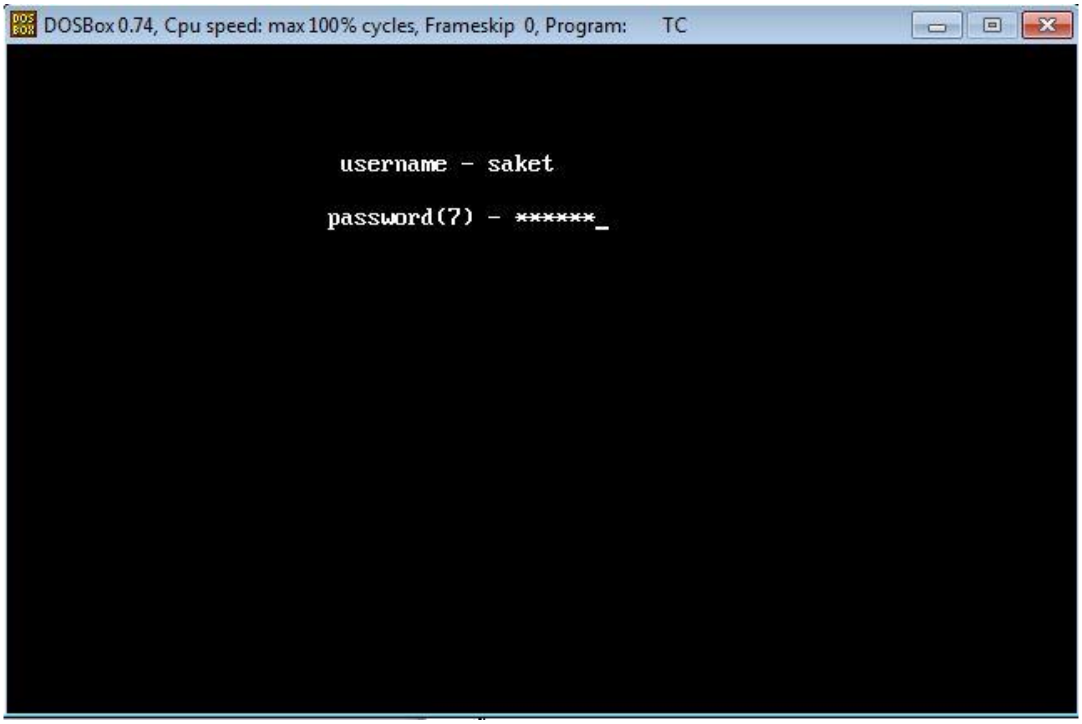
1. Welcome Screen



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

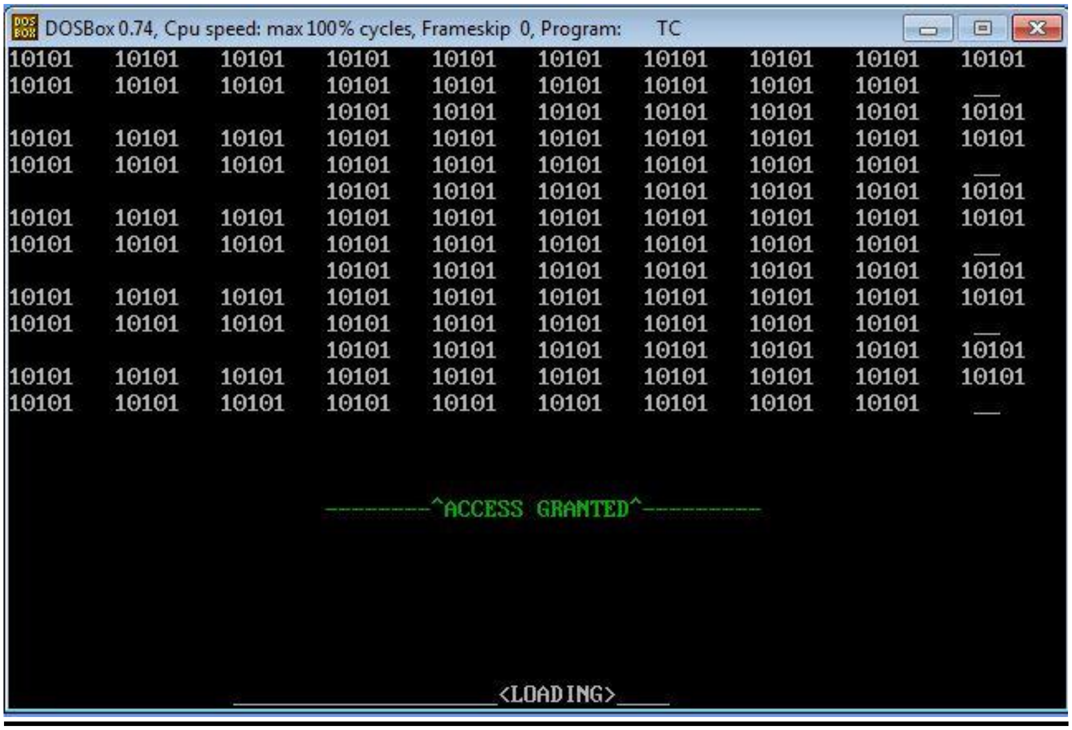
```
You have reached the Database of -


                        ___________________________

                        |         ANONYMOUS       |
                        ___________________________


         We Are Legion. We Do not Forgive. We Do not Forget. Expect Us!

                           Active Since :-  2004..




         Project by - SAKET PANDEY, ARCHISMAN HOTA & MIHIR TIWARI._
```

2. Login Screen


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

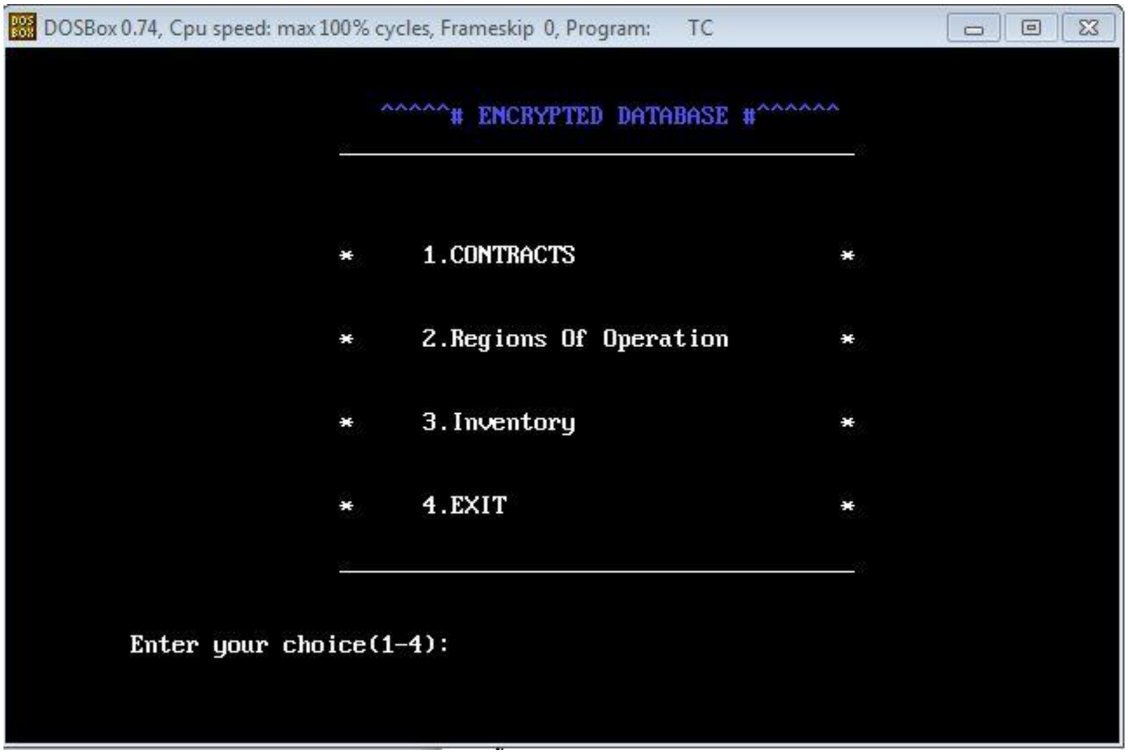
```
username - saket

password(7) - ******_
```

3. Loading Screen


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

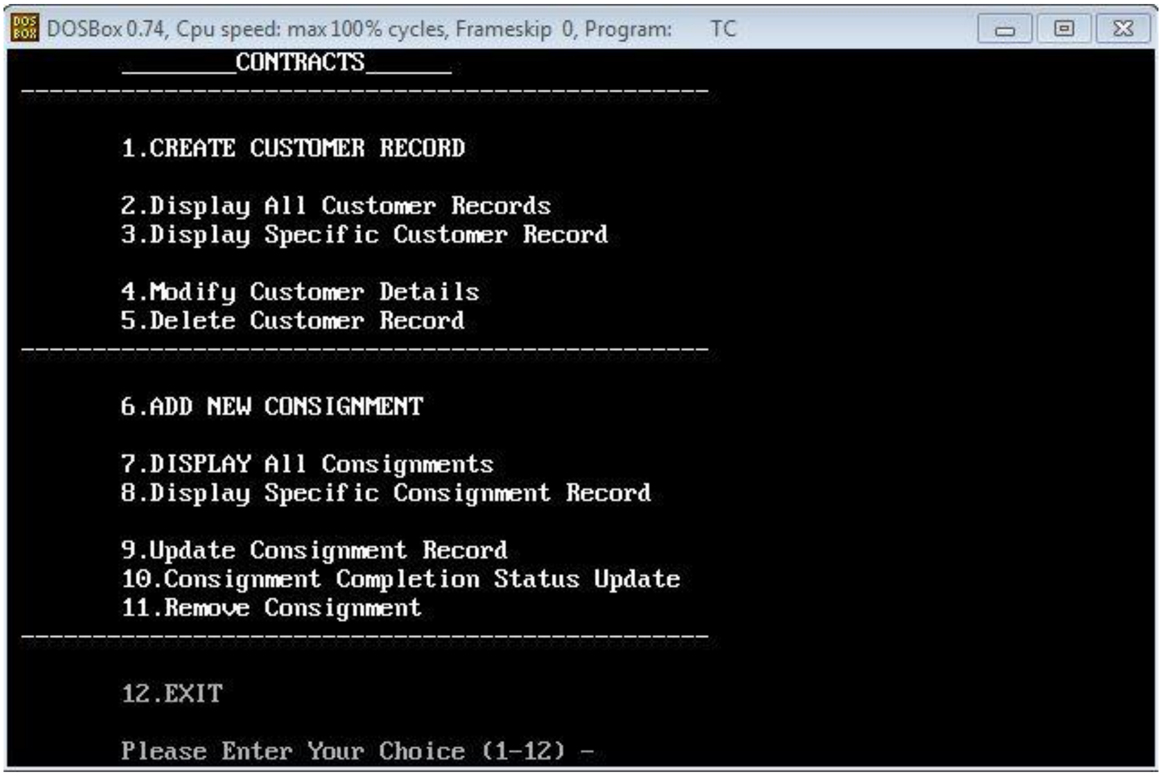
```
10101      10101      10101      10101      10101      10101      10101      10101      10101      10101
10101      10101      10101      10101      10101      10101      10101      10101      10101       __
                                 10101      10101      10101      10101      10101      10101      10101
10101      10101      10101      10101      10101      10101      10101      10101      10101      10101
10101      10101      10101      10101      10101      10101      10101      10101      10101       __
                                 10101      10101      10101      10101      10101      10101      10101
10101      10101      10101      10101      10101      10101      10101      10101      10101      10101
10101      10101      10101      10101      10101      10101      10101      10101      10101       __
                                 10101      10101      10101      10101      10101      10101      10101
10101      10101      10101      10101      10101      10101      10101      10101      10101      10101
10101      10101      10101      10101      10101      10101      10101      10101      10101       __
                                 10101      10101      10101      10101      10101      10101      10101
10101      10101      10101      10101      10101      10101      10101      10101      10101      10101
10101      10101      10101      10101      10101      10101      10101      10101      10101       __



                              ----------^ACCESS GRANTED^-----------




                      _________________________<LOADING>_____
```

4. Main Menu


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

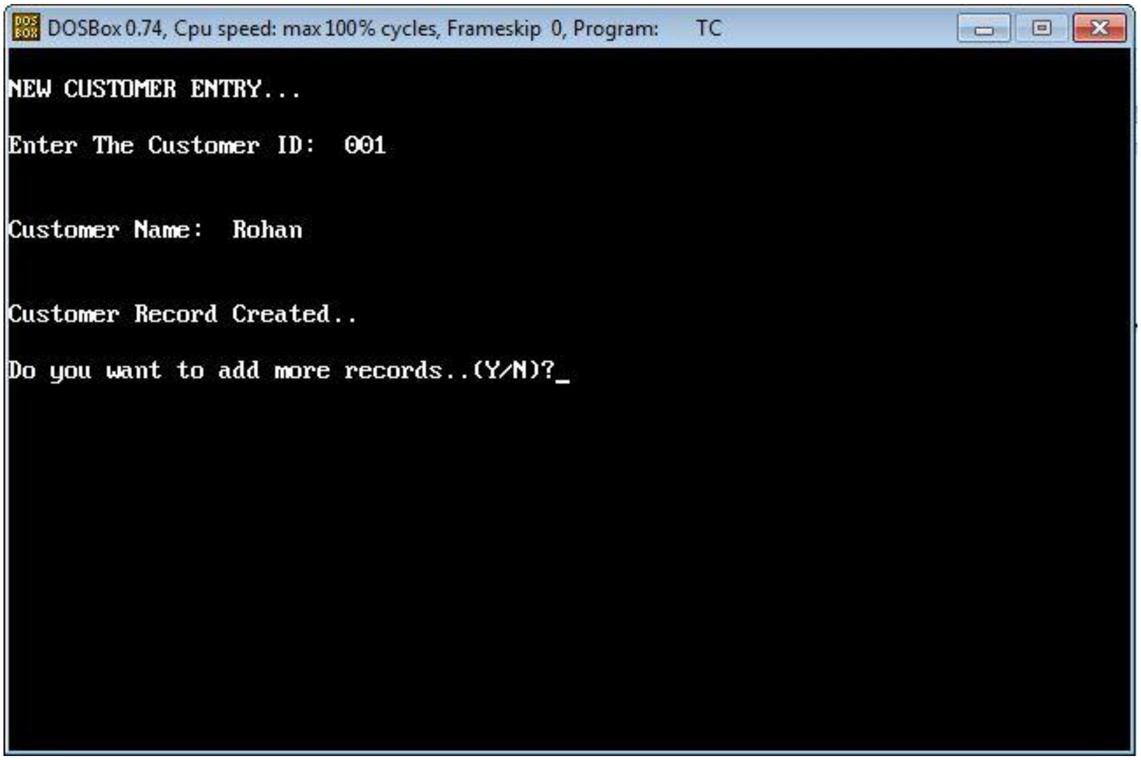
```
                    ^^^^^# ENCRYPTED DATABASE #^^^^^^
                 ____________________________________


                 *     1.CONTRACTS                  *

                 *     2.Regions Of Operation       *

                 *     3.Inventory                  *

                 *     4.EXIT                       *
                 ____________________________________


        Enter your choice(1-4):
```

5. Contracts Menu


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

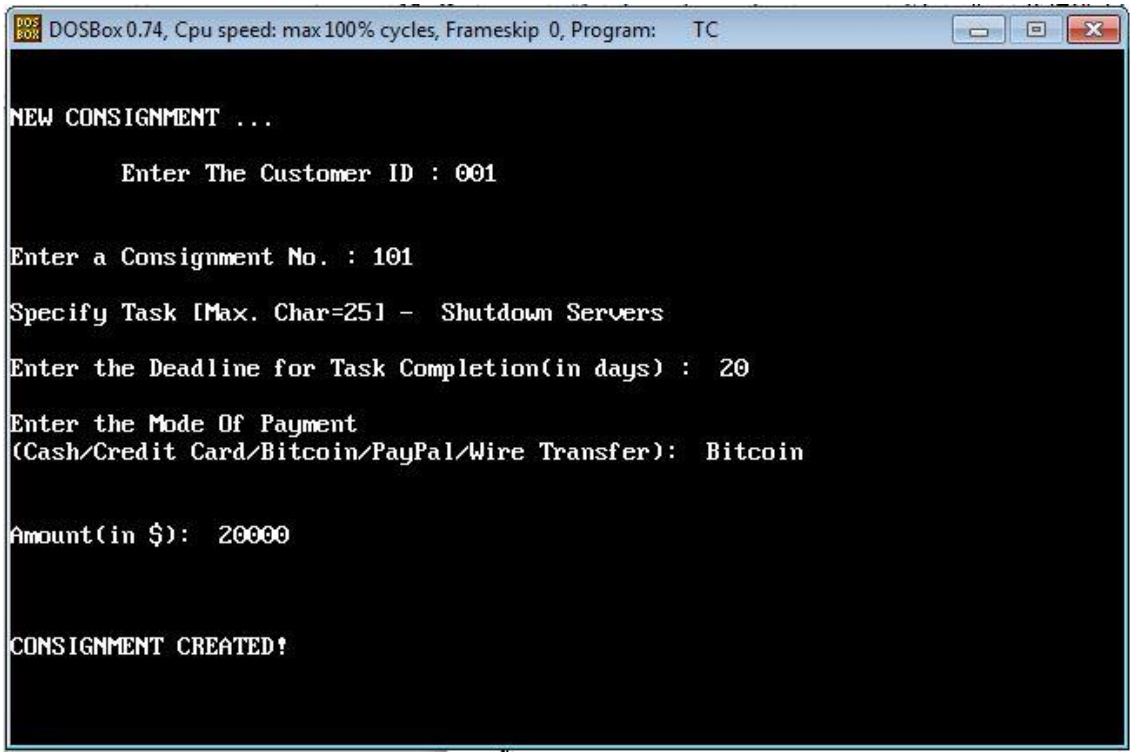
```
       ________CONTRACTS______
------------------------------------------------

       1.CREATE CUSTOMER RECORD

       2.Display All Customer Records
       3.Display Specific Customer Record

       4.Modify Customer Details
       5.Delete Customer Record
------------------------------------------------

       6.ADD NEW CONSIGNMENT

       7.DISPLAY All Consignments
       8.Display Specific Consignment Record

       9.Update Consignment Record
       10.Consignment Completion Status Update
       11.Remove Consignment
------------------------------------------------

       12.EXIT

       Please Enter Your Choice (1-12) -
```

6. New Customer Entry


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

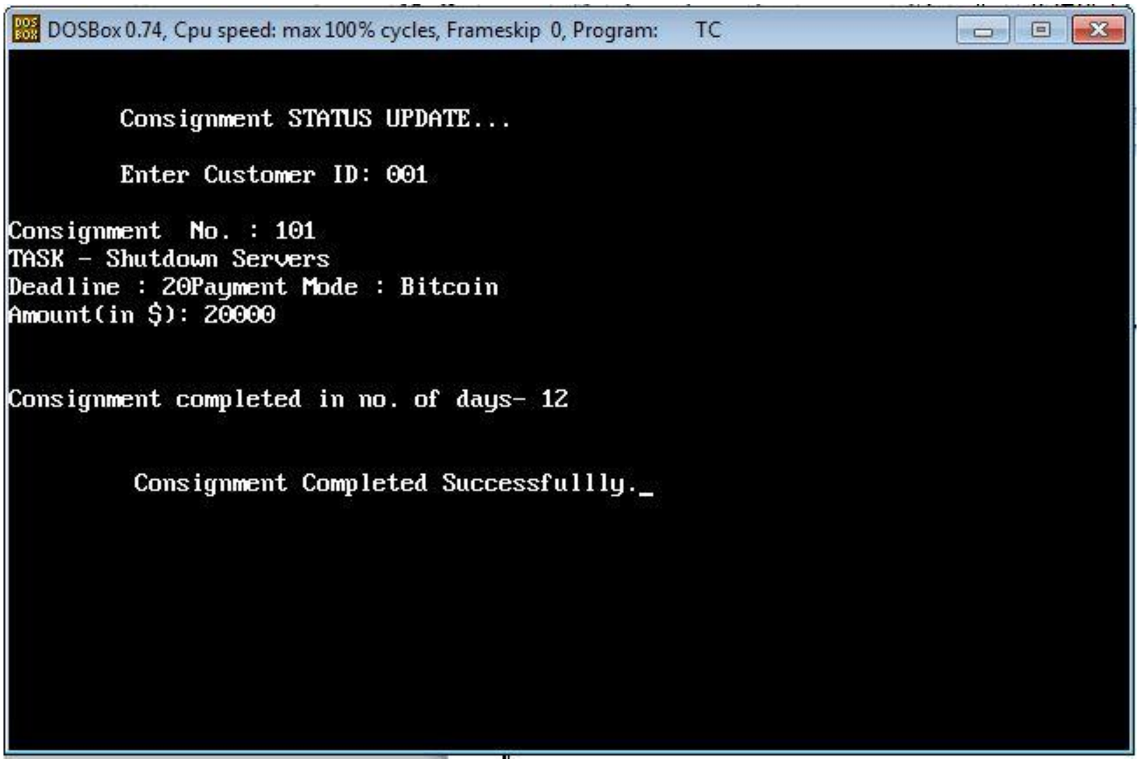
```
NEW CUSTOMER ENTRY...

Enter The Customer ID:  001


Customer Name:  Rohan


Customer Record Created..

Do you want to add more records..(Y/N)?_
```

7. New Consignment Request


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

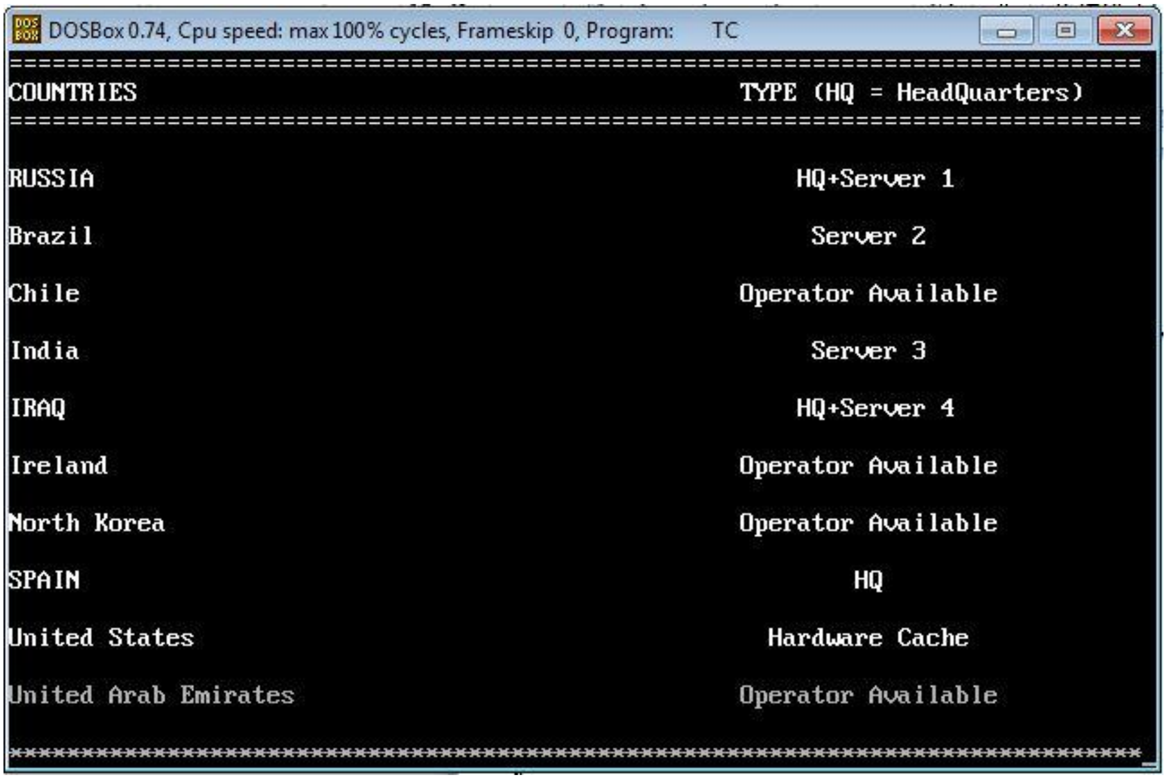
```
NEW CONSIGNMENT ...

        Enter The Customer ID : 001


Enter a Consignment No. : 101

Specify Task [Max. Char=25] -  Shutdown Servers

Enter the Deadline for Task Completion(in days) :  20

Enter the Mode Of Payment
(Cash/Credit Card/Bitcoin/PayPal/Wire Transfer):  Bitcoin


Amount(in $):  20000



CONSIGNMENT CREATED!
```

8. Consignment Status Update


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

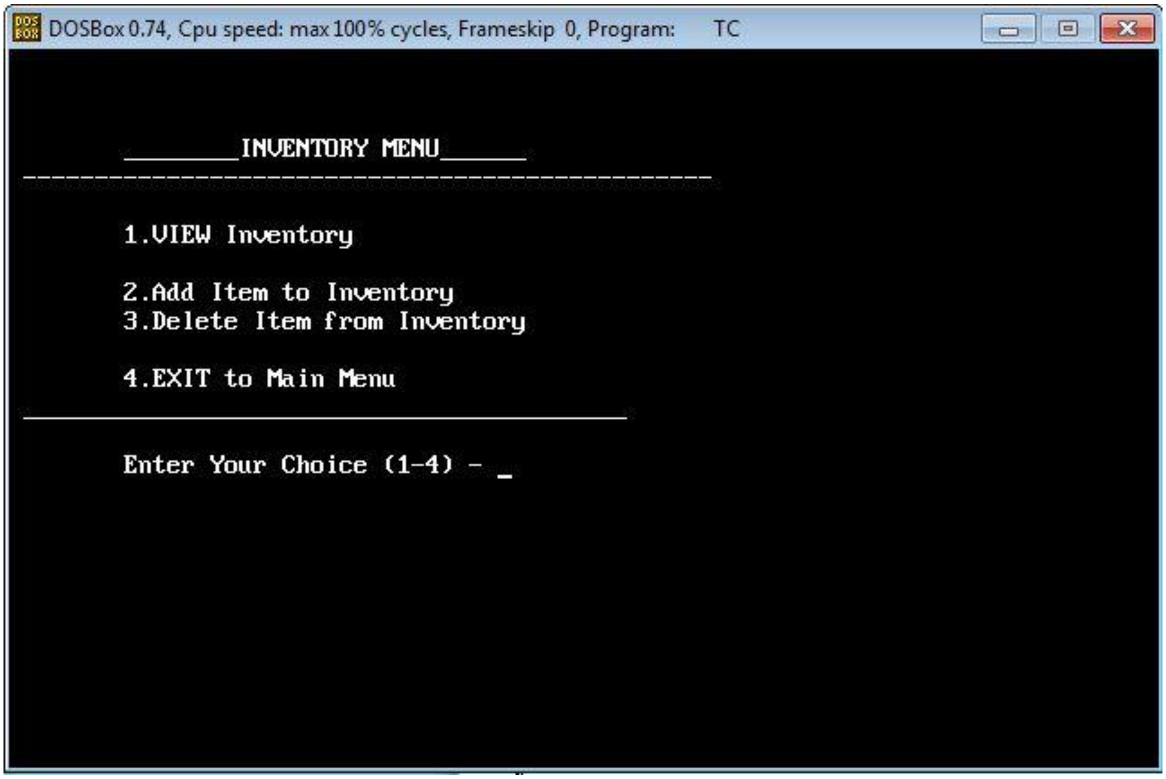
```
        Consignment STATUS UPDATE...

        Enter Customer ID: 001

Consignment  No. : 101
TASK - Shutdown Servers
Deadline : 20Payment Mode : Bitcoin
Amount(in $): 20000


Consignment completed in no. of days- 12


         Consignment Completed Successfullly._
```

9. Regions of Operation


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

```
================================================================================
COUNTRIES                                          TYPE (HQ = HeadQuarters)
================================================================================

RUSSIA                                                 HQ+Server 1

Brazil                                                  Server 2

Chile                                              Operator Available

India                                                   Server 3

IRAQ                                                   HQ+Server 4

Ireland                                            Operator Available

North Korea                                        Operator Available

SPAIN                                                       HQ

United States                                        Hardware Cache

United Arab Emirates                               Operator Available

*******************************************************************************_
```

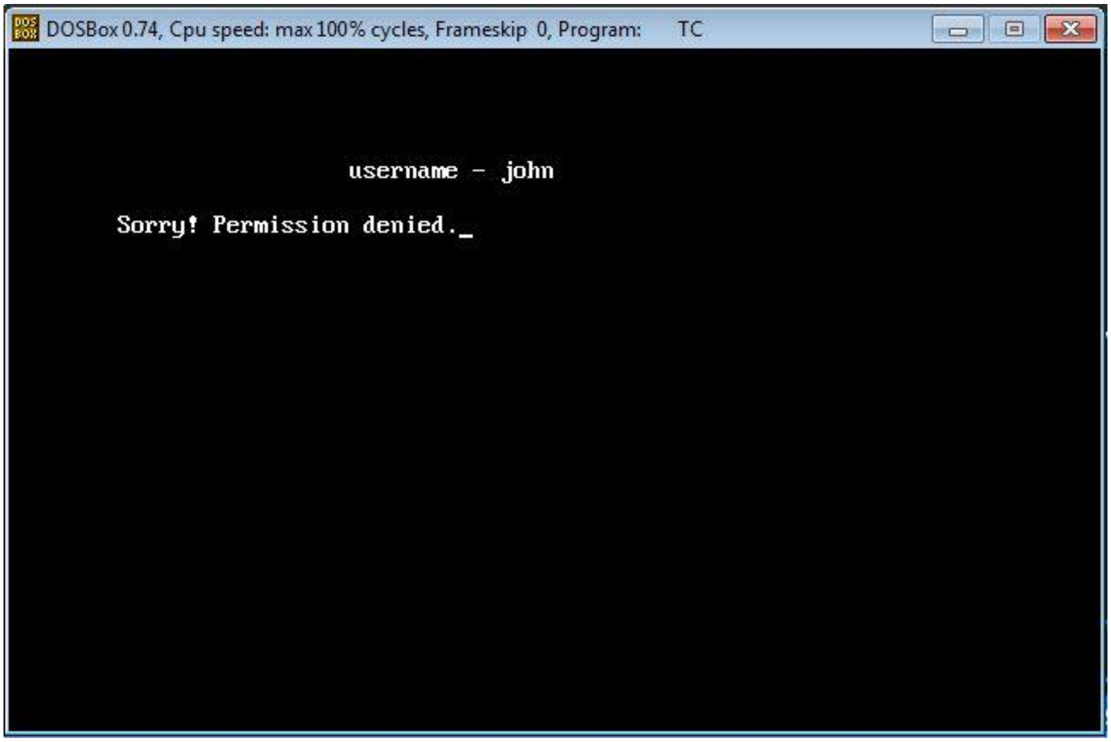
10. Inventory Menu



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

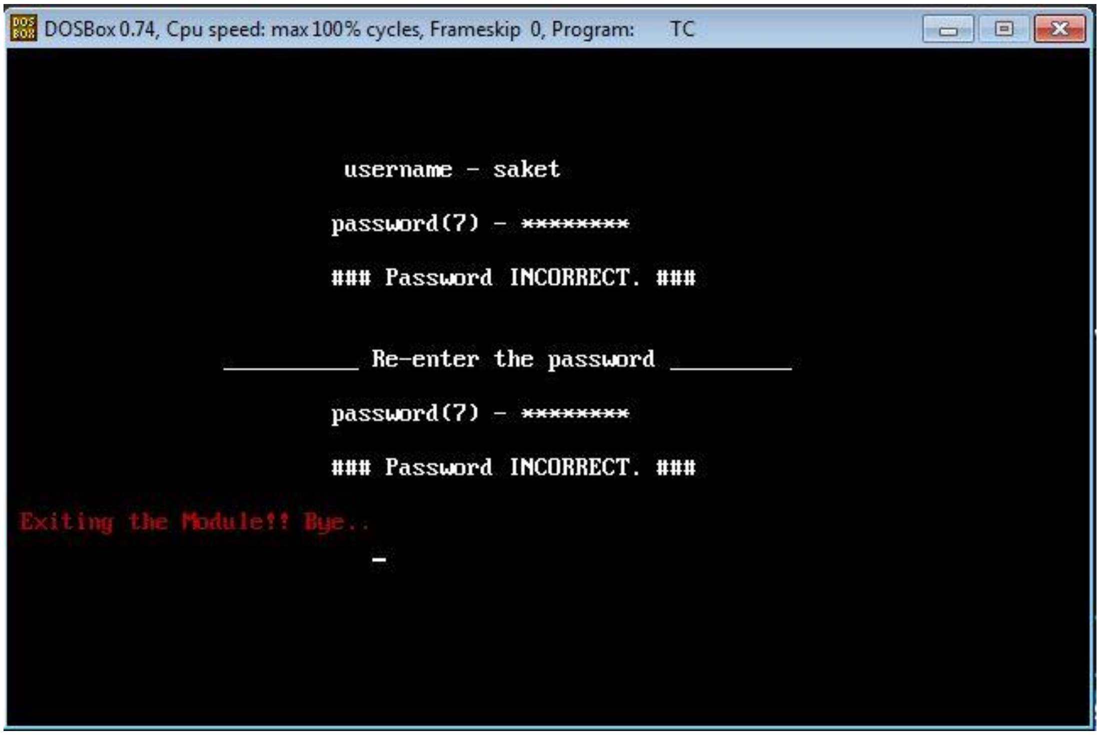
```
        ________INVENTORY MENU______
-----------------------------------------------

        1.VIEW Inventory

        2.Add Item to Inventory
        3.Delete Item from Inventory

        4.EXIT to Main Menu
__________________________________________

        Enter Your Choice (1-4) - _
```

11.Error Screen for Invalid Entries



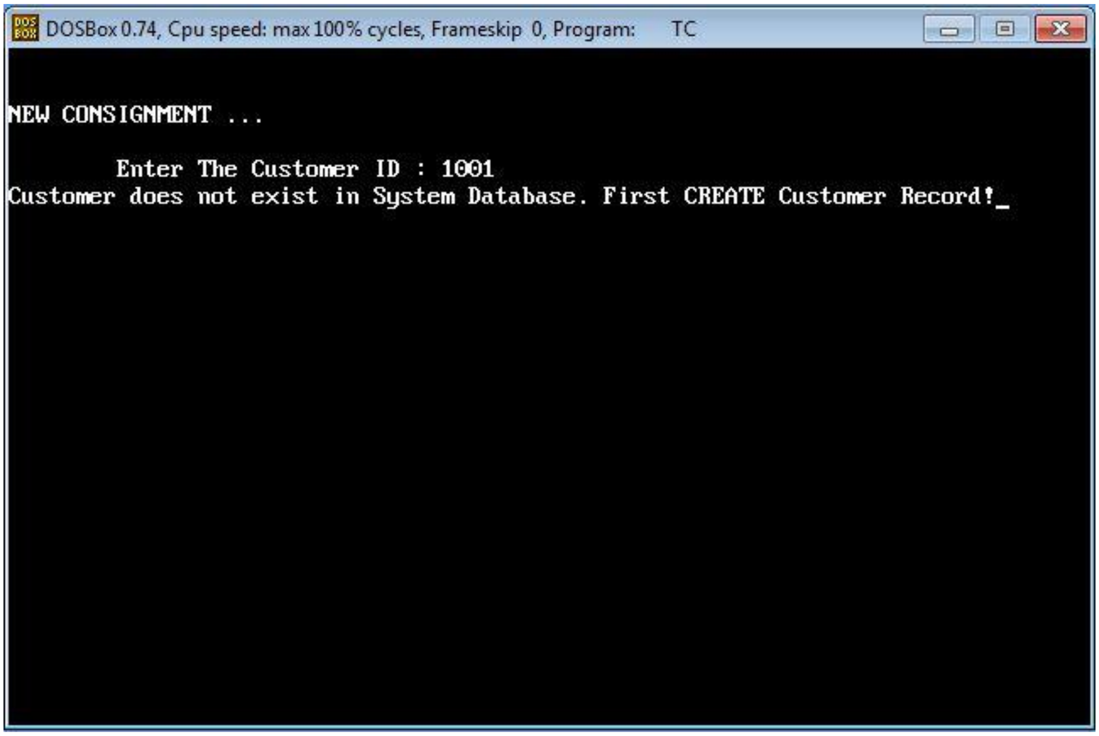
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

```
                        username - john

        Sorry! Permission denied._
```



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

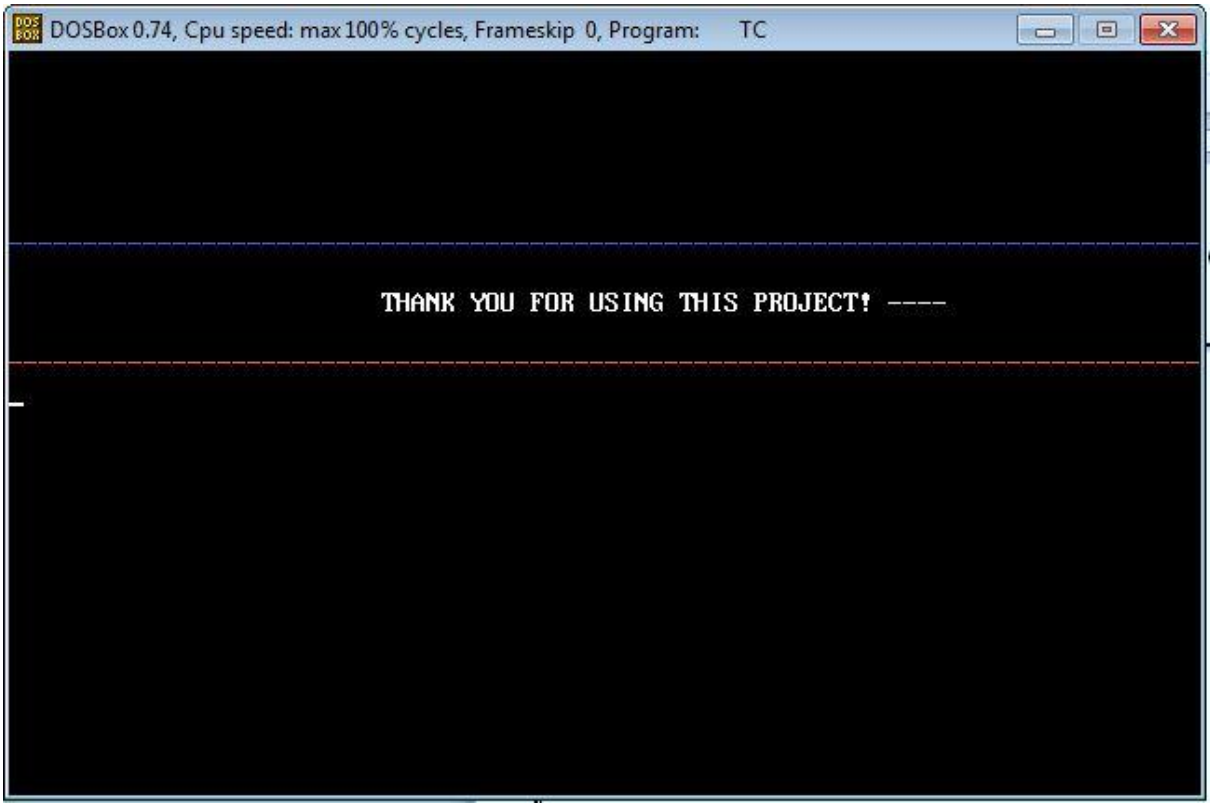
```
                    username - saket

                   password(7) - *********

                   ### Password INCORRECT. ###

             __________ Re-enter the password _________

                   password(7) - *********

                   ### Password INCORRECT. ###

 Exiting the Module!! Bye..
                      _
```



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

```
NEW CONSIGNMENT ...

        Enter The Customer ID : 1001
Customer does not exist in System Database. First CREATE Customer Record!_
```

12.Exit Screen



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

```
THANK YOU FOR USING THIS PROJECT! ----
```

THANK YOU