

```
//Starting of the program
#include<fstream.h>
#include<stdlib.h>
#include<string.h>
#include<conio.h>
#include<conio.h>
#include<ctype.h>
#include<iomanip.h>
#include<stdio.h>
#include<process.h>
#include<math.h>
#include<time.h>
```

```
//Header files
```

```
int admin_or_sales;
admin or salesperson
void admin_password(void);
void sales_password(void);
void sales(void);
void admin(void);
void subscriber_fun();
void payment_fun_admin(void);
void subscriber_create();
void subscriber_modify();
void subscriber_search();
void subscriber_delete();
void subscriber_name_search();
void reqd_details();
void subscriber_no_search();
void subscriber_mobile_search();
void payment_fun_admin();
void payment_view();
void read_details();
void outstanding();
void subscriber_view();
void payment_modify();
void payment_delete();
void pay_search_sub_no();
void pay_search_mobile_no();
void pay_search_bill_no();
void status_set();
void modify_payment();
int mod_or_create;
void topdesign1();
```

```
//To know whether the user is
```

```
class date
{
```

```
    public:
    int m,d;
    long y;
```

```
    void getdate()
```

```
        //to get the date
```

```
    {
        cout<<"\n Enter date: ";
        cin>>d;
        cout<<"\nEnter month(Enter the corresponding number of the
month): ";
        cin>>m;
        cout<<"\n Enter year: ";
        cin>>y;
```

```

    }
    void putdate() // to show the date on the
screen
    {
        cout<<d<<"/"<<m<<"."<<y;
    }
}billdate,paydate,duedate,today;

```

```

int correctdate(date x)
{
    int valid_date;
    valid_date=0;
    if(x.y<2015)
    {
        valid_date=1;
    }
    switch(x.m)
    {
        case 12:
        case 7:
        case 10:
        case 5:
        case 8:
        case 1:
        case 3:if(x.d<1||x.d>31)
            {
                valid_date=1;
            }
            else
            ;
            break;
        case 2:if(x.d<1||x.d>29)
            {
                valid_date=1;
            }
            else
            ;
            break;
        case 4:
        case 6:
        case 9:
        case 11:if(x.d<1||x.d>30)
            {
                valid_date=1;
            }
            else
            ;
            break;
        default:valid_date=1;
            break;
    }
    return valid_date;
};

```

```

int datecmp(date today,date duedate)
{
    int datecheck;
    if(today.y<duedate.y)

```

```

    {
        datecheck=0;
    }
else if(today.y==duedate.y)
{
    if(today.m<duedate.m)
    {
        datecheck=0;
    }
    else if(today.m==duedate.m)
    {
        if(today.d<=duedate.d)
            datecheck=0;
        else
            datecheck=1;
    }
    else
        datecheck=1;
}
else
    datecheck=1;
return datecheck;
}

void topdesign1()                //topdesign1
{
    int i;
    time_t t1=time(NULL);
    tm*t2=localtime(&t1);
    for(i=0;i<79;i++)
    {
        cout<<"=";
    }
    cout<<'\\n';
    for(i=0;i<80;i++)
    {
        cout<<"# ";
    }
    for(i=0;i<79;i++)
    {
        cout<<"=";
    }
    cout<<"\\n\\n\\n";
    cout<<"\\n\\n\\n";
    for(i=0;i<79;i++)
    {
        cout<<"=";
    }
    cout<<'\\n';
    for(i=0;i<80;i++)
    {
        cout<<"# ";
    }
    for(i=0;i<79;i++)
    {
        cout<<"=";
    }
    gotoxy(12,7);
    for(i=0;i<2000;i++)
    {

```

```

        cout<<"D\b";
        if(i==1999)
        {
            cout<<"D ";
        }
    }
    for(i=0;i<2000;i++)
    {
        cout<<"E\b";
        if(i==1999)
        {
            cout<<"E ";
        }
    }
    for(i=0;i<2000;i++)
    {
        cout<<"N\b";
        if(i==1999)
        {
            cout<<"N ";
        }
    }
    for(i=0;i<2000;i++)
    {
        cout<<"D\b";
        if(i==1999)
        {
            cout<<"D ";
        }
    }
    for(i=0;i<2000;i++)
    {
        cout<<"E\b";
        if(i==1999)
        {
            cout<<"E ";
        }
    }
    for(i=0;i<2000;i++)
    {
        cout<<"N\b";
        if(i==1999)
        {
            cout<<"N ";
        }
    }
    for(i=0;i<2000;i++)
    {
        cout<<"M\b";
        if(i==1999)
        {
            cout<<"M ";
        }
    }
    for(i=0;i<2000;i++)
    {
        cout<<"U\b";
        if(i==1999)
        {
            cout<<"U ";
        }
    }

```

```

    }
}
for(i=0;i<2000;i++)
{
    cout<<"S\b";
    if(i==1999)
    {
        cout<<"S ";
    }
}
for(i=0;i<2000;i++)
{
    cout<<"H\b";
    if(i==1999)
    {
        cout<<"H ";
    }
}
for(i=0;i<2000;i++)
{
    cout<<"I\b";
    if(i==1999)
    {
        cout<<"I ";
    }
}
for(i=0;i<2000;i++)
{
    cout<<"V\b";
    if(i==1999)
    {
        cout<<"V ";
    }
}
for(i=0;i<2000;i++)
{
    cout<<"E\b";
    if(i==1999)
    {
        cout<<"E ";
    }
}
for(i=0;i<2000;i++)
{
    cout<<"R\b";
    if(i==1999)
    {
        cout<<"R ";
    }
}
for(i=0;i<2000;i++)
{
    cout<<"S\b";
    if(i==1999)
    {
        cout<<"S ";
    }
}
for(i=0;i<2000;i++)
{

```

```

        cout<<"I\b";
        if(i==1999)
        {
            cout<<"I ";
        }
    }
    for(i=0;i<2000;i++)
    {
        cout<<"O\b";
        if(i==1999)
        {
            cout<<"O ";
        }
    }
    for(i=0;i<2000;i++)
    {
        cout<<"N\b";
        if(i==1999)
        {
            cout<<"N ";
        }
    }
    for(i=0;i<2000;i++)
    {
        cout<<"2\b";
        if(i==1999)
        {
            cout<<"2 ";
        }
    }
    for(i=0;i<2000;i++)
    {
        cout<<".\b";
        if(i==1999)
        {
            cout<<". ";
        }
    }
    for(i=0;i<2000;i++)
    {
        cout<<"0\b";
        if(i==1999)
        {
            cout<<"0 ";
        }
    }
    for(i=0;i<2000;i++)
    {
        cout<<".\b";
        if(i==1999)
        {
            cout<<". ";
        }
    }
    for(i=0;i<2000;i++)
    {
        cout<<"1\b";
        if(i==1999)
        {
            cout<<"1 ";
        }
    }

```

```

    }
    for(i=0;i<2000;i++)
    {
        cout<<"\b";
        if(i==1999)
        {
            cout<<" ";
        }
    }
    for(i=0;i<2000;i++)
    {
        cout<<"5\b";
        if(i==1999)
        {
            cout<<"5 ";
        }
    }
    gotoxy(70,14);
    cout<<t2->tm_mday<<"/"<<(1+t2->tm_mon)<<"/"<<(1900+t2->tm_year);
}

void side()
//sidedesign
{
    int i;
    {
        gotoxy(57,16);
        for(i=0;i<24;i++)
        {
            cout<<"*";
        }
        cout<<'\n';
        gotoxy(57,17);
        {
            cout<<"CONTACT US AT:";
        }
        gotoxy(57,18);
        {
            cout<<"9696969696";
        }
        gotoxy(57,19);
        {
            cout<<"6969696969";
        }
        gotoxy(57,20);
        {
            cout<<"EMAIL US AT:";
        }
        gotoxy(57,21);
        {
            cout<<"dendenmushi@onepiece.com";
        }
        gotoxy(57,22);
        for(i=0;i<24;i++)
        {
            cout<<"*";
        }
    }
}

```

```

void topdesign()                                //topdesign
{
    int i;
    time_t t1=time(NULL);
    tm*t2=localtime(&t1);
    for(i=0;i<79;i++)
    {
        cout<<"=";
    }
    cout<<'\\n';
    for(i=0;i<80;i++)
    {
        cout<<"# ";
    }
    for(i=0;i<79;i++)
    {
        cout<<"=";
    }
    cout<<"\\n\\n\\n";
    cout<<"\\n\\n\\n";
    for(i=0;i<79;i++)
    {
        cout<<"=";
    }
    cout<<'\\n';
    for(i=0;i<80;i++)
    {
        cout<<"# ";
    }
    for(i=0;i<79;i++)
    {
        cout<<"=";
    }
    gotoxy(12,7);
    {
        cout<<"D E N    D E N    M U S H I    V E R S I O N    2.0.1.5";
    }
    gotoxy(70,14);
    cout<<t2->tm_mday<<"/"<<(1+t2->tm_mon)<<"/"<<(1900+t2->tm_year);
}

```

```

int no_of_sub=0;
int s_no=0;

```

```

class subscriber
{
    long int pan_no,credit_no;
    char permanant_address[30],corressponding_address[30];
    char city[10],state[20];
    char email_id[20];
    char scheme[40];
    char facilities_required[50];
    int validity;
    char bill_mode[10],pay_mode[10];
    char status[10];
    int base_tariff;
    unsigned int subscriber_no;
    char mobile_no[11];
}

```



```

char name[20];
char product_type[10];
public:
    void req_details();
    unsigned int ret_sub_no()
    {
        return subscriber_no;
    }
    char* ret_mobile_no()
    {
        return mobile_no;
    }
    char* ret_sub_name()
    {
        return name;
    }
    char* ret_product_type()
    {
        return product_type;
    }
    void input();
    void display();
    void modify_subscriber();
    void ipfacilities();
    void ipbill_mode();
    void ippay_mode();
    void ipstatus();
    void set_status();
    void ippan_no()
    {
        cin>>pan_no;
    }
    void ipcredit_no()
    {
        cin>>credit_no;
    }
    void ippermanent_address()
    {
        gets(permanant_address);
    }
    void ipcorressponding_address()
    {
        gets(corressponding_address);
    }
    void ipcity()
    {
        gets(city);
    }
    void ipstate()
    {
        gets(state);
    }
    void ipemail_id()
    {
        gets(email_id);
    }
    void ipbase_tariff()
    {
        cin>>base_tariff;
    }

```

```

        void ipvalidity()
        {
            cin>>validity;
        }
    }s;

void subscriber::display()                //to display the details
{
    cout<<"\n~~~~~";
    cout<<"\n\t\tName:          "<<name;
    cout<<"\n\t\tSubscriber number:      "<<subscriber_no;
    cout<<"\n\t\tPermanent Address:      "<<permanant_address;
    cout<<"\n\t\tMobile number:          "<<mobile_no;
    cout<<"\n\t\tCorresponding Address: "<<corressponding_address;
    cout<<"\n\t\tPancard number:          "<<pan_no;
    cout<<"\n\t\tState:                  "<<state;
    cout<<"\n\t\tCredit Card Number:     "<<credit_no;
    cout<<"\n\t\tCity:                   "<<city;
    cout<<"\n\t\tScheme:                  "<<scheme;
    cout<<"\n\t\tEmail ID:                 "<<email_id;
    cout<<"\n\t\tValidity:                "<<validity;
    cout<<"\n\t\tProduct type:            "<<product_type;
    cout<<"\n\t\tBill mode:                "<<bill_mode;
    cout<<"\n\t\tPayment mode              "<<pay_mode;
    cout<<"\n\t\tFacilities Required:      "<<facilities_required;
}

void subscriber::ipbill_mode()            //bill receiving mode
{
    int choice5;
    int billmode_check=0;
    do
    {
        billmode_check=0;
        cout<<"\n Enter bill receiving mode: ";
        cout<<"\n1)Email";
        cout<<"\n2)Post";
        cout<<"\n3)SMS\n";
        cin>>choice5;
        switch(choice5)
        {
            case 1:strcpy(bill_mode,"Email");
                    break;
            case 2:strcpy(bill_mode,"Post");
                    break;
            case 3:strcpy(bill_mode,"SMS");
                    break;
            default:cout<<"\nEnter correct option";
                    billmode_check=1;
                    break;
        }
    }
    while(billmode_check==1);
}

void subscriber::ipstatus()              //status

```

```

{
    int status_check=0;
    int choice3;
    do
    {
        cout<<"\nEnter status(1-2):";
        cout<<"\n1.Active:";
        cout<<"\n2.Inacive";
        cin>>choice3;
        switch(choice3)
        {
            case 1:strcpy(status,"Active");
                    break;
            case 2:strcpy(status,"Inactive");
                    break;
            default:cout<<"\nInvalid Option";
                    break;
        }
    }
    while(status_check==1);
}

void subscriber::ippay_mode()                //mode of paying the bill
{
    int paymode_check;
    int choice2;
    do
    {
        paymode_check=0;
        cout<<"\nEnter paymode(1-3):";
        cout<<"\n1) Internet";
        cout<<"\n2) Cash";
        cout<<"\n3) Cheque";
        cin>>choice2;
        switch(choice2)
        {
            case 1:strcpy(pay_mode,"Internet");
                    break;
            case 2:strcpy(pay_mode,"Cash");
                    break;
            case 3:strcpy(pay_mode,"Cheques");
                    break;
            default:cout<<"\n Invalid Option";
                    paymode_check=1;
                    break;
        }
    }
    while(paymode_check==1);
}

void subscriber::ipfacilities()              //facilities that are
available
{
    int fac_reqd;
    int p,q,r=0,vidcalling=0,conf=0,sms=0;
    char morefac;
    do
    {

```

```

p=0,q=0;
cout<<"\nEnter facilites required:";
cout<<"\n1)Video Calling";
cout<<"\n2)Conference";
cout<<"\n3)SMS";
cin>>fac_reqd;
switch(fac_reqd)
{
    case 1:if(vidcalling==0)
        {
            strcpy(facilities_required,"video
Calling");
            vidcalling=1;
            r++;
        }
    else
        cout<<"\nFacility already added";
        break;
    case 2:if(conf==0)
        {
            strcpy(facilities_required,"Conference");
            conf=1;
            r++;
        }
    else
        cout<<"\nFacilities already added";
        break;
    case 3:if(sms==0)
        {
            strcpy(facilities_required,"SMS");
            r++;
            sms=1;
        }
    else
        cout<<"\nFacilities already added";
        break;
}
cout<<"\nDo you want more facilities??(y/n)\n";
cin>>morefac;
if(morefac=='y')
{
    strcat(facilities_required,",");
    q=1;
}
}
while((p==1||q==1)&&(r<4));
}

```

```

int check_mobile_no(char* mob_no)
{
    int flag;
    subscriber d;
    flag=0;
    ifstream mobilecheck("subscribe.dat",ios::binary);
    while(mobilecheck.read((char*)&d,sizeof(d)))
    {
        if(strcmp(d.ret_mobile_no(),mob_no)==0)
        {

```

```

        flag=1;
    }
}
return flag;
}
int mobile_no_check(char* mob_no)
{
    int flag=0,mob_length;
    mob_length=strlen(mob_no);
    if(mob_length==10)
    {
        for(int i=0;i<10;i++)
        {if(!isdigit(mob_no[i]))
        {flag=1;
        }
        }
    }
    else
    flag=1;
    return flag;
}
void subscriber::input()    //subscriber input
{
    int choice,choice1,choice2,choice3,choice5;
    clrscr();
    topdesign();
    cout<<"\n enter subscriber name:-";
    gets(name);
    subscriber_no=rand();
    cout<<"subscriber number :-"<<subscriber_no;
    int mobcheck=0,checkmob=0;
    mob:
    cout<<"\n enter mobile no:-";
    gets(mobile_no);
    mobcheck=check_mobile_no(mobile_no);
    if(mobcheck==1)
    {cout<<"\nmobile number already exists\n";
    goto mob;
    }
    checkmob=mobile_no_check(mobile_no);
    if (checkmob==1)
    {
        cout<<"\n mobile number must have 10 numbers ";
        checkmob=1;
        goto mob;
    }
    int pro_check; //product type check
    do
    {cout<<"\n enter product type(1-2):- \n 1.prepaid\n 2.postpaid";
    cin>>choice;
    pro_check=0;
    if(choice==1)
    strcpy(product_type,"prepaid");
    else if(choice==2)
        strcpy(product_type,"postpaid");
    else
    {
        cout<<"\n invalid option";
        pro_check=1;
    }
}

```

```

}while(pro_check==1);
cout<<"\n enter pan no:-";
cin>>pan_no;
cout<<"\n enter scheme(1-3)";
cout<<"\n 1.Internet\n 2.message scheme\n 3.talking scheme";
cin>>choicel;
if(choicel==1) //scheme check
strcpy(scheme,"internet");
else if(choicel==2)
strcpy(scheme,"message");
else if(choicel==3)
strcpy(scheme,"talking");
cout<<"\n enter credit card no";
cin>>credit_no;
cout<<"\n enter permanent address";
gets(permanant_address);
cout<<"\n enter corressponding address";
gets(corressponding_address);
cout<<"\n enter city name :-";
gets(city);
cout<<"\n enter state name";gets(state);
cout<<"\n enter email id";
gets(email_id);
int fac_reqd;
int p,q,r=0,vidcalling=0,conf=0,sms=0;
char morefac;
do{
p=0,q=0;
cout<<"\n enter facilities required:\n1.video
calling\n2.conference\n3.sms";
cin>>fac_reqd;
switch(fac_reqd) //matching the facility
{
case 1: if(vidcalling==0)
        { strcpy(facilities_required,"video calling");
          vidcalling=1;
          r++;
        }
        else
          cout<<"\n facility already added";
          break;
case 2: if(conf==0)
        { strcpy(facilities_required,"conference");
          conf=1;
          r++;
        }
        else cout<<"\n facility already added";
          break;
case 3: if(sms==0)
        { strcpy(facilities_required,"sms");
          r++;
          sms=1;
        }
        else cout<<"\n facility already added";
          break;
default: clrscr();
        topdesign();
        cout<<"\n enter correct choice";
        p=1;
        break;
}
}

```

```

}
cout<<"\n do you want more facilities (Y/N)\n";
cin>>morefac;
if(morefac=='y')
{ strcat(facilities_required,",");
  q=1;
}

}while((p==1||q==1)&&(r,4));
cout<<"\n enter validity:-";
cin>>validity;
int billmode_check;
do{
billmode_check=0;
cout<<"\n enter bill receiving mode:- \n 1.email \n 2.post \n 3.sms \n";
cin>>choice5;
switch(choice5)
{
case 1: strcpy(bill_mode,"email");
        break;
case 2: strcpy(bill_mode,"post");
        break;
case 3: strcpy(bill_mode,"sms");
        break;
default:cout<<"\n enter correct option";
        billmode_check=1;
        break;
}
}while(billmode_check==1);
int paymode_check;
do
{
paymode_check=0;
cout<<"\n enter paymode(1-3):- \n 1.internet \n 2.cash \n 3.cheque";
cin>>choice2;
switch(choice2)
{
case 1: strcpy(pay_mode,"internet");
        break;
case 2: strcpy(pay_mode,"cash");
        break;
case 3: strcpy(pay_mode,"cheque");
        break;
default: cout<<"\n invalid option";
        paymode_check=1;
        break;
}
}while(paymode_check==1);
int status_check;
do
{
status_check=0;
cout<<"\n enter status:- \n 1. active \n 2. inactive ";
cin>>choice3;
switch(choice3)
{
case 1: strcpy(status,"active");
        break;
case 2: strcpy(status,"inactive");
        break;

```

```

default: cout<<"\n invalid option ";
           status_check=1;
           break;
}
}while(status_check==1);
cout<<"\n enter base tariff:";
cin>>base_tariff;
}
void login()      //to login
{
clrscr();
topdesign1();
side();
status_set();
int mchoice;
log:
gotoxy(25,16);
cout<<"\n login menu"<<"\n\n\t\t\t";
cout<<"1.administrator"<<"\n\t\t\t";
cout<<"2.saleperson\n\n";
gotoxy(26,21);
cin>>mchoice;
if(mchoice==1)
admin_password();
else if(mchoice==2)
sales_password();
else
{cout<<"\n wrong choice";
goto log;
}
}
void admin_password() //admin password
{
clrscr();
topdesign();
int pass_check=0;
char change;
char password1[20];
char password[]={"tele@#\$"};
char username1[20];
char username[]={"admin"};
user:
    clrscr();
    topdesign();
    cout<<setw(40)<<"username:-";
    gets(username1);
    if(strcmp(username1,username)==0)
    {
pass:
        clrscr();
        topdesign();
        cout<<setw(40)<<"username:-"<<username1;
        cout<<'\\n'<<setw(37)<<"password:-";
        for(int i=0;i<20;i++)
        {password1[i]=getch();
        int a=password1[i];
        if(a==8)
        {if(i>0)
        { cout<<'\\b';
        i-=2;

```



```

        }
        if(i== -1)
            goto pass;
    }
    if(password1[i]=='\r')
    {
        password1[i]='\0';
        if(strcmp(password,password1)==0)
            admin();
        else
        {cout<<'\\n'<<setw(40)<<"incorrect password";
        pass_check++;
        if (pass_check<=3)
        {cout<<'\\n'<<"          press enter";
            getch();
            goto pass;
        }
        else
            login();
    }
}
}
else if(a==8)
;
else
    putchar('*');
}
}
else
{ clrscr();
topdesign();
cout<<setw(30)<<"\\nincorrect username";
cout<<"do you want to change user?? (y/n)";
cin>>change;
if(change=='y' || change=='Y')
    login();
else
    goto user;
}
}

void sales_password()    //sales password
{
    clrscr();
    topdesign();
    int pass_check=0;
    char change;
    char password1[20];
    char password[]={"tele@#\$"};
    char username1[20];
    char username[]={"sales"};
    user:
        clrscr();
        topdesign();
        cout<<setw(40)<<"username:-";
        gets(username1);
        if(strcmp(username1,username)==0)
        {
            pass:
                clrscr();
                topdesign();
                cout<<setw(40)<<"username:-"<<username1;

```

```

        cout<<'\\n'<<setw(37)<<"password:-";
        for(int i=0;i<20;i++)
        {password1[i]=getch();
         int a=password1[i];
         if(a==8)
         {if(i>0)
          { cout<<'\\b';
           i-=2;
          }
          if(i== -1)
          goto pass;
         }
        if(password1[i]=='\\r')
        {
            password1[i]='\\0';
            if(strcmp(password,password1)==0)
            sales();
            else
            {cout<<'\\n'<<setw(40)<<"incorrect password";
             pass_check++;
             if (pass_check<=3)
             {cout<<'\\n'<<"          press enter";
              getch();
              goto pass;
             }
            else
            sales();
           }
        }
        else if(a==8)
        ;
        else
        putchar('*');
    }
}
else
{ clrscr();
topdesign();
cout<<setw(30)<<"\\nincorrect username";
cout<<"do you want to change user?? (y/n)";
cin>>change;
if(change=='y' || change=='Y')
login();
else
goto user;
}
}

```

```

void admin() //admin
{clrscr();
topdesign();
cout<<"\\n\\t\\t\\t Welcome Admin ";
admin_or_sales=1;
int flag=1;
int choose;
do
{
cout<<"\\n\\n"<<setw(44)<<"1.subscriber
functions"<<'\\n'<<setw(40)<<"2.payment

```

```

functions"<<'\\n'<<setw(38)<<"3.generate
report"<<'\\n'<<setw(30)<<"4.logout\\n";
cin>>choose;
switch(choose)
{ case 1: subscriber_fun();
        flag=0;
        break;
    case 2: payment_fun_admin();
        flag=0;
        break;
    case 3: reqd_details();
        flag=0;
        break;
    case 4: login();
        break;
    default: cout<<"eneter proper choice";
        flag=1;
        break;
}
}while(flag==1);
}
void subscriber_fun()                //subscriber functions
{
int z;
do
{
    clrscr();
    topdesign();
    cout<<'\\n'<<setw(40)<<"subscriber options";
    cout<<"\\n\\n"<<setw(40)<<"1.create subscriber";
    cout<<'\\n'<<setw(40)<<"2.modify subscriber";
    cout<<'\\n'<<setw(40)<<"3.delete subscriber";
    cout<<'\\n'<<setw(40)<<"4.search subscriber";
    cout<<'\\n'<<setw(46)<<"5.view subscriber details";
    cout<<'\\n'<<setw(27)<<" 6.back";
    cin>>z;
    switch(z)
    { case 1:subscriber_create();
        break;
        case 2:subscriber_modify();
        break;
        case 3:subscriber_delete();
        break;
        case 4:subscriber_search();
        break;
        case 5:subscriber_view();
        break;
        case 6:admin();
        break;
        default : cout<<"\\n wrong option";
            z=1;
            break;
    }
}while(z==1);
}

void subscriber_view()                //to view the subscriber
details
{
int flag=0;

```

```

clrscr();
topdesign();
ifstream view("subscriberd.dat",ios::binary);
while(view.read((char*)&s,sizeof(s)))
{flag=1;
  s.display();
}
if(flag==0)
cout<<"\n no records";
cout<<"\n press enter to go back";
getche();
if(admin_or_sales==1)
subscriber_fun();
else
sales();
}
void subscriber_create() //create new
subscriber
{
char ch;
subscriber create;
ofstream subcreate;
subcreate.open("subscriberd.dat",ios::binary|ios::app);
if(!subcreate)
{cerr<<"\n co";
exit(1);
}
do
{
create.input();
subcreate.write((char*)&create,sizeof(create));
cout<<"do you want to add more subscribers?";
cin>>ch;
}while(ch=='Y' || ch=='y');
subcreate.close();
}
void subscriber_modify() //subscriber modify
{
int found=0;
subscriber s2;
long int pos;
ifstream f1("subscriberd.dat",ios::binary);
ofstream f2("temp.dat",ios::binary||ios::app);
unsigned int sno;
cout<<"\n enter the subscriber number you want to modify";
cin>>sno;
while(!f1.eof())
{
pos=f1.tellg();
f1.read((char*)&s2,sizeof(s2));
if(sno==s2.ret_sub_no())
{
s2.modify_subscriber();
f2.seekp(pos);
f2.write((char*)&s2,sizeof(s2));
found=1;
break;
}
else
f2.write((char*)&s2,sizeof(s2));
}
}

```

```

}
f1.close();
f2.close();
remove("subscriberd.dat");
rename("temp.dat","subscriberd.dat");
if(found==0)
{
cout<<"\n record not found";
cout<<"\n press any key to go to subscribers";
getche();
subscriber_fun();
}
else
{
cout<<"\n\nrecord sucessfully modified";
cout<<"\n\npressenter to go to admin menu";
getche();
subscriber_fun();
}
}
void subscriber::modify_subscriber()
{
clrscr();
topdesign();
int mobcheck;
int checkmob=0;
int mod_ch;
cout<<"\n\t\t\t 1.name\n\t\t\t 2. mobile number";
cout<<"\n\t\t\t 3.product type \n\t\t\t 4. pan number";
cout<<"\n\t\t\t 5.credit card number \n\t\t\t 6. permanent address";
cout<<"\n\t\t\t 7.corressponding address \n\t\t\t 8. city";
cout<<"\n\t\t\t 9. state\n\t\t\t 10.email id";
cout<<"\n\t\t\t 11. facilities required\n\t\t\t 12. validity";
cout<<"\n\t\t\t 13. billmode\n\t\t\t 14. payment mode";
cout<<"\n\t\t\t 15 status\n\t\t\t 16.base tariff";
cout<<"\n\t\t\t enter no to be modified";
cin>>mod_ch;
int mod_prototype;
switch(mod_ch)
{
case 1: cout<<"\n enter the new name";
        gets(name);
        break;
case 2: mob:
        cout<<"\n enter mobile number:-";
        gets(mobile_no);
        mobcheck=check_mobile_no(mobile_no);
        if(mobcheck==1)
        {
            cout<<"\nmobile number already exists\n";
            goto mob;
        }
        checkmob=mobile_no_check(mobile_no);
        if(checkmob==1)
        {
            cout<<"\n mobile number must have 10 numbers only";
            checkmob=1;
            goto mob;
        }
        break;

```

```

case 3 :cout<<"enter subscription type";
        cout<<"\n 1.prepaid";
        cout<<"\n 2.postpaid";
        cout<<"\n enter your choice(1-2):-";
        cin>>mod_prototype;
        switch(mod_prototype)
        {
            case 1: strcpy(product_type,"prepaid");
                    break;
            case 2: strcpy(product_type,"postpaid");
                    break;
            default:cout<<"\ninvalid option";
                    break;
        }
        break;
case 4:cout<<"\n enter your new pan card no:-";
        cin>>pan_no;
        break;
case 5:cout<<"\n enter your new credit card no:-";
        cin>>credit_no;
        break;
case 6:cout<<"\n enter your new permanent address:-";
        gets(permanant_address);
        break;
case 7:cout<<"\n enter your new corressponding address:-";
        gets(corressponding_address);
        break;
case 8:cout<<"\nenter your new city:-";
        gets(city);
        break;
case 9:cout<<"\n enter your new state";
        gets(state);
        break;
case 10:cout<<"\n enter your new email id:-";
        gets(email_id);
        break;
case 11:s.ipfacilities();
        break;
case 12:cout<<"\n enter your new validity period:-";
        cin>>validity;
        break;
case 13:s.ipbill_mode();
        break;
case 14:s.ippay_mode();
        break;
case 15:cout<<"\n enter your new base tariff:-";
        cin>>base_tariff;
        break;
default :cout<<"\nwrong option";
        cout<<"\npres enter to go to subscriber menu";
        getch();
        subscriber_fun();
        break;
}
}
void subscriber_search() //subscriber search
{
    int search_option;
    clrscr();

```

```

topdesign();
cout<<"\nsearch:-";
cout<<"\n1.name\n2.subscribnumber\n3.mobile number";
cin>>search_option;
switch(search_option)
{
case 1: subscriber_name_search();
        break;
case 2: subscriber_no_search();
        break;
case 3: subscriber_mobile_search();
        break;
default: cout<<"\n wrong option";
        cout<<"\n press enter to tgo to subscriber menu";
        getch();
        subscriber_fun();
        break;
}
}

void subscriber_name_search() //Search using
subscriber name
{
    int flag=0;
    subscriber name_search;
    char sub_name[20];
    cout<<"\nEnter subscriber name to search";
    gets(sub_name);
    ifstream sns("subscriberd.dat",ios::binary);
    if(!sns)
    {
        cerr<<"cannot open file";
        exit(1);
    }
    while(sns.read((char*)&name_search,sizeof(name_search)))
    {
        if (strcmp(sub_name,name_search.ret_sub_name())==0)
        {
            flag=1;
            name_search.display();
        }
    }
    if(flag==0)
    {
        cout<<"\nSubscriber not found";
    }
    sns.close();
    cout<<"\nPress enter to go to subscriber menu";
    getch();
    subscriber_fun();
}

void subscriber_no_search() //Search using the
subscriber's number
{
    int flag=0;
    subscriber no_search;
    int sub_no;
    cout<<"\nEnter subscriber number to be searched";
    cin>>sub_no;
    ifstream snos("subscriberd.dat",ios::binary);

```

```

if(!snos)
{
    cerr<<"Cannot open file";
    exit(1);
}
while(snos.read((char*)&no_search,sizeof(no_search)))
{
    if(no_search.ret_sub_no()==sub_no)
    {
        no_search.display();
        flag=1;
        snos.close();
    }
}
if(flag==0)
{
    cout<<"\nSubscriber number not found";
}

cout<<"\nPress enter to go to subscriber menu";
getch();
subscriber_fun();
}
void subscriber_mobile_search() //Search using the
mobile number
{
    int checkmob;
    int flag=0;
    subscriber mobile_search;
    char mobno[11];
    mob:
    cout<<"\nEnter mobile number:";
    gets(mobno);

    checkmob=mobile_no_check(mobno);
    if(checkmob==1)
    {
        cout<<"\nMobile number must have 10 numbers only";
        checkmob=1;
        goto mob;
    }

    ifstream sms("subscriberd.dat",ios::binary);
    if(!sms)
    {
        cerr<<"Cannot open file";
        exit(1);
    }
    while(sms.read((char*)&mobile_search,sizeof(mobile_search)))
    {
        if(strcmp(mobile_search.ret_mobile_no(),mobno)==0)
        {
            mobile_search.display();
            flag=1;
            sms.close();
        }
    }
    if(flag==0)
    {
        cout<<"\nSubscriber name not found";
    }
}

```



```

    }
    cout<<"\nPress enter to go to subscriber menu";
    getch();
    subscriber_fun();
}
void subscriber_delete() //Deleting the
subscriber
{
    int sno;
    int achoice=2;
    subscriber s3;
    ifstream f3("subscriberd.dat",ios::binary);
    ofstream f4("temp.dat",ios::binary);

    int flag=0;
    cout<<"\nEnter the subscriber number to be deleted:";
    cin>>sno;
    while(f3.read((char*)&s3,sizeof(s3)))
    {
        if(s3.ret_sub_no()==sno)
        {
            flag=1;
            s3.display();
            cout<<"\nAre you sure you want to delete this subscriber....if yes
press1 else press 2";
            cin>>achoice;
            if(achoice==2)
                f4.write((char*)&s3,sizeof(s3));
            else
            {
                ;
            }
        }
        else
            f4.write((char*)&s3,sizeof(s3));
    }
    if(flag==0)
    {
        cout<<"\nSubscriber not found";
    }
    f3.close();
    f4.close();
    remove("subscriberd.dat");
    rename("temp.dat","subscriberd.dat");
    cout<<"\nPress enter";
    getch();
    subscriber_fun();
}

```

```

class payment
{
    char product_type[10];
    int subscriber_no;
    char name[20];
    int bill_no;
    char mobile_no[11];
    date billing_date;
    date pay_date;
    date due_date;
    float recharge;
}

```

```

int total_bill;
int amt_paid;
int amt_due;
char trans_status[20];
public:
void display();
date ret_bill_date()
{
    return billing_date;
}
date ret_due_date()
{
    return due_date;
}
void set_billdate(date date)
{
    billing_date=date;
}
char* ret_product_type_p()
{
    return product_type;
}
char* ret_name_p()
{
    return name;
}

int ret_sub_no_p()
{
    return subscriber_no;
}
char* ret_mobile_no_p()
{
    return mobile_no;
}
int ret_bill_no()
{
    return bill_no;
}
char* ret_trans_status()
{
    return trans_status;
}
void payment_input();
void modify_payment();
void set_sub_no(int number)
{
    subscriber_no=number;
}
void set_name(char *subname,int length);
void set_product_type(char *type,int length);
void set_mobile_no(char* number)
{
    strcpy(mobile_no,number);
}
};
void payment::set_name(char *subname,int length)
{
    for(int i=0;i<length;i++)
        name[i]=*(subname+i);
}

```

```

    name[i]='\0';
}
void payment::set_product_type(char *type,int length)
{
    for(int i=0;i<length;i++)
        product_type[i]=*(type+i);
    product_type[i]='\0';
}

void subscriber::req_details()
{

cout<<"\n\n"<<subscriber_no<<setw(10)<<name<<setw(20)<<mobile_no<<setw(15)
)<<status<<setw(10)<<scheme;
}

void reqd_details()                //To display the report
{
    subscriber a;
    ifstream report;
    clrscr();
    topdesign();

    cout<<"\nSUBNO"<<setw(10)<<"SUBNAME"<<setw(20)<<"MOBILENO"<<setw(15)<<"ST
ATUS"<<setw(10)<<"SCHEME";
    report.open("subscriberd.dat",ios::binary);
    while(report.read((char*)&a,sizeof(a)))
    {
        a.req_details();
    }
    report.close();
    cout<<"\n\n\nPress enter";
    getche();
    if(admin_or_sales==1)
        admin();
    else
        sales();
}

void payment::modify_payment()                //To modify the payment
details
{
    clrscr();
    topdesign();
    int fine=0;
    int paycheck1=0;int paycheck=0;
    int billcheck1=0;
    int duecheck,duecheck1=0;
    int choice;
    if(strcmpi(product_type,"Prepaid")==0)
    {cout<<"\n\t\t\t\t\tModify menu:";
    cout<<"\n\t\t\t\t\t1.billdate:";
    cout<<"\n\t\t\t\t\t2.Recharge amount";
    cout<<"\n\t\t\t\t\t3.Back";
    cin>>choice;

    switch(choice)
    {
        case 1:
            do
            {

```

```

        billcheck1=0;
        cout<<"\nBilldate:";
        billdate.getdate();
        billing_date=billdate;
        int cdate=correctdate(billing_date);
        if(cdate==1)
        {
            cout<<"\nWrong date";
            billcheck1=1;
            cout<<"\nPress enter to re-enter date";
            getche();
        }
        else
            billcheck1=0;
    }while(billcheck1==1);
    break;

case 2:
    cout<<"\nEnter amount recharged";
    cin>>recharge;
    amt_paid=0;
    total_bill=0;
    amt_due=0;
    break;

case 3:if(admin_or_sales==1)
    payment_fun_admin();
    else
        sales();
    break;

default:
    cout<<"\nWrong option";
    cout<<"\nPress enter to continue";
    getche();
    if(admin_or_sales==1)
        payment_fun_admin();
    else
        sales();
    break;
}
}
else
{
    cout<<"\n\t\t\t\t\tModify menu:";
    cout<<"\n\n\t\t\t\t\t1.Bill date:";
    cout<<"\n\t\t\t\t\t2.Due date:";
    cout<<"\n\t\t\t\t\t3.Pay date:";
    cout<<"\n\t\t\t\t\t4.bill amount";
    cout<<"\n\t\t\t\t\t5.Back";
    cin>>choice;
    switch(choice)
    {
        case1:
            do
            {
                billcheck1=0;
                cout<<"\nBill date:";
                billdate.getdate();
                billing_date=billdate;
                int cdate=correctdate(billing_date);
                if(cdate==1)
                {

```

```

        cout<<"\nWrong date";
        billcheck1=1;
        cout<<"\nPress enter to re-nter the
date";

        getch();
    }
    else
        billcheck1=0;
    }while(billcheck1==1);
    break;

case2:
do
{
    duecheck=0;
    duecheck1=0;
    cout<<"\nDue date:";
    duedate.getdate();
    due_date=duedate;

    int ddate=correctdate(due_date);
    if(ddate==1)
    {
        cout<<"\nWrong date";
        duecheck=1;
        cout<<"\nPress enter to re-enter date";
        getch();
    }
    else
        duecheck=0;
    if(datecmp(billing_date,due_date)==1)
    {
        cout<<"\nDue date cant be before bill date";
        cout<<"\nPress enter to re-enter";
        getch();
        duecheck1=1;
    }
    else
        duecheck1=0;
    }while(duecheck==1||duecheck1==1);

break;

case3:                                pay:

do

{ paycheck=0;

    paycheck1=0;

    cout<<"\nDate paid:";

    paydate.getdate();

    pay_date=paydate;

    int pdate=correctdate(pay_date);

    if(pdate==1)

    {

```

```

        cout<<"\nWrong date";

        paycheck=1;

        cout<<"\nPress enter to re-entr the date";

        getch();
    }
    else
    paycheck=0;

        if(datecmp(billing_date,pay_date)==1)
    {
        cout<<"\nDate cant be before bill date";

        cout<<"\nPress enter to re-enter";

        getch();

        paycheck1=1;
    }
    else
    paycheck1=0;

    }while(paycheck==1||paycheck1==1);

    break;

        case4:

if(datecmp(due_date,pay_date)==1)
    {
        fine=1;
    }

        cout<<"\nEnter total bill";

        cin>>total_bill;

        if(fine==1)
        {
            cout<<"\nFine:"<<(20/100)*total_bill;

            total_bill=total_bill+(20/100)*total_bill;
        }

```

```

        cout<<"\nEnter amount paid";

        cin>>amt_paid;

        amt_due=total_bill-amt_paid;

        recharge=0;

        break;

case5:          if(admin_or_sales==1)

                payment_fun_admin();
                else

                                                                sales();

                                                                break;

default:

        cout<<"\nWrong option";

        cout<<"\nPress enter to continue";

        getch();

        if(admin_or_sales==1)

                payment_fun_admin();

                else

                        sales();

                break;
                }
                if(amt_due==0)
                {
                        strcpy(trans_status,"paid");
                }
                else if(amt_due>0)
                {
                        strcpy(trans_status,"outstanding");
                }
                }
        }

void payment::payment_input() //To input the payment details
{
        int fine;
        bill_no=rand();
        cout<<"\nBill number:"<<bill_no;
        int billcheck=0;
        if(strcmp(product_type,"Prepaid")==0)
        {
                do
                {
                        billcheck=0;

```

```

cout<<"\nBill date:";
billdate.getdate();
billing_date=billdate;
int cdate=correctdate(billing_date);
if(cdate==1)
{
    cout<<"\nWrong date";
    getche();
}
else
billcheck=0;
}while(billcheck==1);
    cout<<"\nEnter amount recharged";
    cin>>recharge;
    amt_paid=0;
    total_bill=0;
    amt_due=0;
}
else
{
    do
    {
        billcheck=0;
        cout<<"\nBill date:";
        billdate.getdate();
        billing_date=billdate;
        int cdate=correctdate(billing_date);
        if(cdate==1)
        {
            cout<<"\nWrong date";
            billcheck=1;
            cout<<"\nPress enter to re-enter date";
            getche();
        }
        else
        billcheck=0;
    }while(billcheck==1);
    due:
    int duecheck=0;
    int duecheck1=0;
    do
    {
        duecheck=0;
        duecheck1=0;
        cout<<"\nDue date:";
        duedate.getdate();
        due_date=duedate;

        int ddate=correctdate(due_date);
        if(ddate==1)
        {
            cout<<"\nWrong date";
            duecheck=1;
            cout<<"\nPress enter to re-enter date";
            getche();
        }
        else
        duecheck=0;
        if(datecmp(billing_date,duedate)==1)
        {

```



```

        cout<<"\nDue date cant be before bill date";
        cout<<"\nPress enter to re-enter";
        getch();
        duecheck1=1;
    }
    else
        duecheck1=0;
}while(duecheck==1||duecheck1==1);
pay:
int paycheck=0,paycheck1=0;
do
{
    paycheck=0;
    paycheck1=0;
    cout<<"\nDate paid:";
    paydate.getdate();
    pay_date=paydate;
    int pdate=correctdate(pay_date);
    if(pdate==1)
    {
        cout<<"\nWrong date";
        paycheck=1;
        cout<<"\nPress enter to re-enter date";
        getch();
    }
    else
        paycheck=0;
    if(datecmp(billing_date,pay_date)==1)
    {
        cout<<"Date paid cant be before bill date";
        cout<<"\nPress enter to re-enter";
        getch();
        paycheck1=1;
    }

    else
        paycheck1=0;
    if(datecmp(due_date,pay_date)==1)
    {
        fine=1;
    }
}while(paycheck==1||paycheck1==1);
cout<<"\nEnter total bill";
cin>>total_bill;
cout<<"\nEnter amount paid";
cin>>amt_paid;
if(fine==1)
{
    cout<<"\nFine:"<<(20/100)*total_bill;
    total_bill=total_bill+(20/100)*total_bill;
}
amt_due=total_bill-amt_paid;
recharge=0;
}
if(amt_due==0)
{
    strcpy(trans_status,"paid");
}
else if(amt_due>0)
{

```

```

        strcpy(trans_status,"outstanding");
    }
}

void payment::display()          //To display the payment details of the
subscriber
{

cout<<"\n~~~~~";
    cout<<"\n\n\t\tName:          "<<name;
    cout<<"\n\n\t\tSubscriber number:      "<<subscriber_no;
    cout<<"\n\t\t\tProduct type:          "<<product_type;
    cout<<"\n\t\t\tMobile no:            "<<mobile_no;
    cout<<"\n\t\t\tBill no:              "<<bill_no;
    if(strcmp(product_type,"prepaid")==0)
    {
        cout<<"\n\t\t\tBill date:
"<<billing_date.d<<"."<<billing_date.m<<"."<<billing_date.y;
        cout<<"\n\t\t\tRecharge amount:      "<<recharge;
        cout<<"\n\t\t\tTransaction status:    "<<trans_status;
    }
    else
    {
        cout<<"\n\t\t\tDue date:
"<<due_date.d<<"."<<due_date.m<<"."<<due_date.y;
        cout<<"\n\t\t\tDate paid:
"<<pay_date.d<<"."<<pay_date.m<<"."<<pay_date.y;
        cout<<"\n\t\t\tTotal bill:      "<<total_bill;
        cout<<"\n\t\t\tAmount paid:      "<<amt_paid;
        cout<<"\n\t\t\tAmount due:      "<<amt_due;
        cout<<"\n\t\t\tTransaction status:    "<<trans_status;
    }
}

void payment_create()          //to create a new payment
{
paycreated:
clrscr();
topdesign();
int flag=0;
int subno_c;
char contin='n';
subscriber s1;
payment creation;
ofstream paycreate("paymnt.dat",ios::binary|ios::app);
ifstream sub("subscriberd.dat",ios::binary);
flag=0;
contin='n';
cout<<"\n enter subscriber number to create payment\n";
cin>>subno_c;
sub.seekg(0);
while(sub.read((char*)&s1,sizeof(s1)))
{
if(s1.ret_sub_no()==subno_c)
{
creation.set_sub_no(s1.ret_sub_no());
creation.set_product_type(s1.ret_product_type(),strlen(s1.ret_product_type()));
creation.set_name(s1.ret_sub_name(),strlen(s1.ret_sub_name()));
creation.set_mobile_no(s1.ret_mobile_no());

```

```

creation.payment_input();
paycreate.write((char*)&creation,sizeof(creation));
flag=1;
}
}
paycreate.close();
sub.close();
if(flag==0)
{
cout<<"\n record not found";
getche();
if(admin_or_sales==1)
payment_fun_admin();
else
sales();
}
else
{
cout<<"\n record created\npress enter";
getche();
cout<<"\n want to create more payments?(y/n) ";
cin>>contin;
if(contin=='y' || contin=='Y')
{
goto paycreated;
}
else
{
if(admin_or_sales==1)
payment_fun_admin();
else
sales();
}
}
}
void payment_fun_admin() //payment functions
{
clrscr();
topdesign();
int option2;
int flag=0;
int option1;
do
{
clrscr();
topdesign();
flag=0;
cout<<"\n                payment options:-\n";
cout<<"\n                1.create payment";
cout<<"\n                2.view payment details";
cout<<"\n                3.modify payment";
cout<<"\n                4.delete payment details";
cout<<"\n                5.search";
cout<<"\n                6.view outstanding bills";
cout<<"\n                7.back\n";
cin>>option1;
switch(option1)
{
case 1:cout<<"\n payment creation";
        payment_create();

```

```

        break;
case 2:cout<<"\n view payment details";
        payment_view();
        break;
case 3:cout<<"\n modification of payment details";
        payment_modify();
        break;
case 4:cout<<"\n deletion of payment details";
        payment_delete();
        break;
case 5:
        cout<<"\n search options";
        cout<<"\n1.subscriber number";
        cout<<"\n2.mobile number";
        cout<<"\n3.bill number\n";
        cin>>option2;
        switch(option2)
        {
            case 1: pay_search_sub_no();
                    break;
            case 2: pay_search_mobile_no();
                    break;
            case 3: pay_search_bill_no();
                    break;
            default: cout<<"\n enter proper choice";
                    flag=1;
                    break;
        }
        break;
case 6: outstanding();
        break;
case 7: admin();
        break;
default: cout<<"\n enter proper option";
        cout<<"\n press enter to continue";
        getche();
        flag=1;
    }
}while(flag==1);
}

```

```

void payment_view()          //to view the payment
{
    clrscr();
    topdesign();
    int bno;
    int found=0;
    payment p;
    ifstream view;
    view.open("paymnt.dat",ios::binary);
    while(view.read((char*)&p,sizeof(p)))
    {
        p.display();
    }
    view.close();
    cout<<"\n\npress enter to go back";
    getche();
    if(admin_or_sales==1)
    payment_fun_admin();
}

```

```

else
sales();
}
void payment_modify()    //to modify the payment details
{
clrscr();
topdesign();
int bno;
int found=0;
payment p;
ifstream paymod("paymnt.dat",ios::binary);
ofstream of("temp.dat",ios::binary|ios::app);
cout<<"\n enter the bill number of bill you want to modify";
cin>>bno;//pay mod
while(paymod.read((char*)&p,sizeof(p)))
{
if(bno==p.ret_bill_no())
{
p.set_sub_no(p.ret_sub_no_p());
p.set_product_type(p.ret_product_type_p(),strlen(p.ret_product_type_p()));
;
p.set_mobile_no(p.ret_mobile_no_p());
p.set_name(p.ret_name_p(),strlen(p.ret_name_p()));
p.modify_payment();
of.write((char*)&p,sizeof(p));
found=1;
}
else
of.write((char*)&p,sizeof(p));
}
of.close();
paymod.close();
remove("paymnt.dat");
rename("temp.dat","paymnt.dat");
if(found==0)
{
cout<<"\n record not found";
cout<<"\n press any any key to go back";
getche();
if(admin_or_sales==1)
payment_fun_admin();
else
sales();
}
else{
cout<<"\n record sucessfully modified";
cout<<"\n press enter to got to admin menu";
getche();
if(admin_or_sales==1)
payment_fun_admin();
else
sales();
}
}

```

```

void pay_search_sub_no() //searching payment using subscriber number
{
clrscr();
topdesign();

```

```

int flag=0;
payment no_search;
int sub_no;
cout<<"\n enter subscriber number to be searched:-";
cin>>sub_no;
ifstream psnos("paymnt.dat",ios::binary);
if(!psnos)
{
cerr<<"cannot open file";
exit(1);
}
while(psnos.read((char*)&no_search,sizeof(no_search))) // searching for
pay
{
if(no_search.ret_sub_no_p()==sub_no)
{
no_search.display();
flag=1;
psnos.close();
}
}
if(flag==0)
{
cout<<"\n subscriber not found";
}
cout<<"\n press enter to go to the payment menu";
getch();
if(admin_or_sales==1)
payment_fun_admin();
else
sales();
}
void pay_search_mobile_no() //payment search using mobile number
{
clrscr();
topdesign();
int flag=0;
payment mobile_search;
char mobno[11];
int checkmob=0;
mob:
cout<<"\n enter mobile number:-";
gets(mobno);
checkmob=mobile_no_check(mobno);
if(checkmob==1)
{
cout<<"\n mobile number must have 10 digits only";
checkmob=1;
goto mob;
}
ifstream psms("paymnt.dat",ios::binary);
if(!psms)
{
cerr<<"cannot open file";
exit(1);
}
while(psms.read((char*)&mobile_search,sizeof(mobile_search)))
{
if(strcmp(mobile_search.ret_mobile_no_p(),mobno)==0)
{

```

```

mobile_search.display();
flag=1;
psms.close();
}
}
if(flag==0)
{
cout<<"\n Mobile number not found";
}
cout<<"\n press enter to go to payment menu";
getch();
if(admin_or_sales==1)
payment_fun_admin();
else
sales();
}
void pay_search_bill_no() // searching payment using bill number
{
clrscr();
topdesign();
int flag=0;
payment bill_search;
int billno;
cout<<"\n enter bill number to be searched";
cin>>billno;
ifstream psbs("paymnt.dat",ios::binary);
if(!psbs)
{
cerr<<"\n cannot open file";
exit(1);
}
while(psbs.read((char*)&bill_search,sizeof(bill_search)))
{
if(bill_search.ret_bill_no()==billno)
{
bill_search.display();
flag=1;
psbs.close();
}
}
if(flag==0)
{
cout<<"\n bill number not found";
}
cout<<"\n press enter to go to the payment menu";
getche();
if(admin_or_sales==1)
payment_fun_admin();
else
sales();
}
void payment_delete() // to delete payment
{
clrscr();
topdesign();
int bno;
int achoice=2;
payment pay_delete;
ifstream f3("paymnt.dat");
ofstream f4("temp.dat",ios::binary);

```

```

int flag=0;
cout<<"\n enter the bill number of the payment to be deleted:-";
cin>>bno;
while(f3.read((char*)&pay_delete,sizeof(pay_delete)))
{
if(pay_delete.ret_bill_no()==bno)
{
flag=1;    //pay delete
pay_delete.display();
cout<<"\n are you sure you want to delete payment of this subscriber?";
cout<<"\n 1.yes";
cout<<"\n 2.no";
cin>>achoice;
if(achoice==2)
f4.write((char*)&pay_delete,sizeof(pay_delete));
else
;
}
else
f4.write((char*)&pay_delete,sizeof(pay_delete));
}
f3.close();
f4.close();
if(flag==0)
{
cout<<"\n subscriber not found";
}
remove("paymnt.dat");
rename("temp.dat","paymnt.dat");
cout<<"\n press enter to go to the payment menu";
getche();
if(admin_or_sales==1)
payment_fun_admin();
else
sales();
}
void sales()    //function for salesperson
{
clrscr();
topdesign();
int searchcheck;
int pays;
admin_or_sales=2;
int sale_check=0;
int sale_option;
clrscr();
topdesign();
cout<<"\n\t\t\t Welcome salesperson :)";
do
{
sale_check=0;
cout<<"\n"<<"                payment options:-\n";
cout<<"\n\n"<<"                1.create payment";
cout<<"\n"<<"                2.modify payment";
cout<<"\n"<<"                3.search";
cout<<"\n"<<"                4.view payments";
cout<<"\n"<<"                5.view subscriber details";
cout<<"\n"<<"                6.view outstanding bills";
cout<<"\n"<<"                7.logout";
cin>>sale_option;

```



```

switch(sale_option)
{
case 1:
payment_create();
break;
case 2:
payment_modify();
break;
case 3:
do
{
clrscr();
topdesign();
cout<<"\n"<<setw(35)<<"search options\n";
cout<<"\n"<<setw(35)<<"1. subscriber number";
cout<<"\n"<<setw(35)<<"2. mobile number";
cout<<"\n"<<setw(35)<<"3. bill number\n";
cin>>pays;
switch(pays)
{
case 1:
pay_search_sub_no();
break;
case 2:
pay_search_mobile_no();
break;
case 3:
pay_search_bill_no();
break;
default:
cout<<"\n enter proper choice";
searchcheck=1;
break;
}
}while(searchcheck==1);
break;
case 4:
payment_view();
break;
case 5:
subscriber_view();
break;
case 6:
outstanding();
break;
case 7:
login();
break;
default:
cout<<"\n enter the correct option";
sale_check=1;
break;
}
}while(sale_check==1);
}
void outstanding()          //to view the outstanding bill
{
clrscr();
topdesign();
payment dp;

```

```

ifstream outtrans;
outtrans.open("paymnt.dat",ios::binary);
if(!outtrans)
{
cerr<<"\n cannot open file";
exit(1);
}
cout<<"\n payment detail with outstanding bills";
while(outtrans.read((char*)&dp,sizeof(dp)))
{if(strcmp(dp.ret_trans_status(),"outstanding")==0)
{
dp.display();
}
}
outtrans.close();
cout<<"\n press enter to go back";
getche();
if(admin_or_sales==1)
admin();
else
sales();
}
void status_set()
{
subscriber ss;
ifstream fi("subscriberd.dat",ios::binary);
ofstream fo("temporary.dat",ios::binary);
if(!fi)
{
;
}
while(fi.read((char*)&ss,sizeof(ss)))
{
ss.set_status();
fo.write((char*)&ss,sizeof(ss));
}
remove("subscriberd.dat");
rename("temporary.dat","subscriberd.dat");
}
int get_status(tm*t2,date comp)
{
int compdate;
if(comp.y<(1900+t2->tm_year))
{
compdate=0;
}
else if(comp.y==(1900+t2->tm_year))
{
if(comp.m<(1+t2->tm_mon))
{
compdate=0;
}
else if(comp.m==(1+t2->tm_mon))
{
if(comp.d<t2->tm_mday)
compdate=0;
else
compdate=1;
}
else

```

```

compdate=1;
}
else
compdate=1;
return compdate;
}
void subscriber::set_status()    // to set the status of the subscriber
{
time_t t1=time(NULL);
tm *t2=localtime(&t1);
payment dcheck;
int dateverify;
dateverify=get_status(t2,dcheck.ret_due_date());
if(dateverify==1)
{
strcpy(status,"inactive");
}
}
void main()
{
unsigned int seed;
time_t t;
seed=(unsigned)time(&t);
srand(seed);
login();
}

```