OBJECT ORIENTED PROGRAMMING PROJECT ON-

Medicine Distribution Database

Made By:

B116003	ADHESH GARG
B116036	PRATEEK AGRAWAL
B116039	PRATYUSH SAMANTARAY
B116051	SONAKSHI SAXENA
B516057	HARDIK SINGH

INDEX

Page No.

- Objective 2.
- Concepts Used 2.
- Class Diagrams:
  - ADMIN Class 3.
  - MEDICINE Class 4.
  - MEDICINESHOP Class 5.
  - DISTRICT Class 6.
- Snapshots of Interfaces 7.
- Snapshots of Backend File Systems 9.
- Source Code 15.
- References 36.

Objective

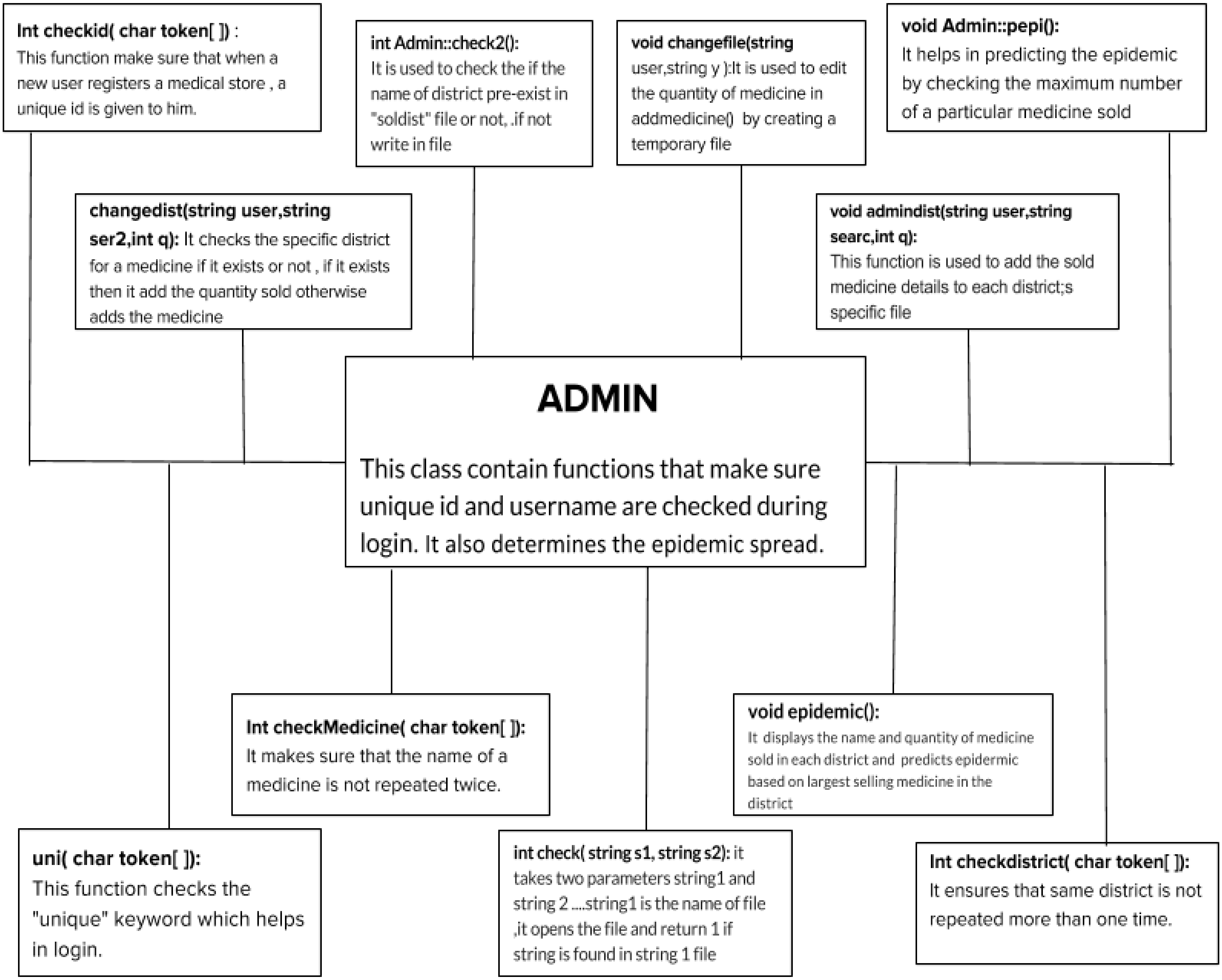
To create a database for drugs and medicine distribution for Odisha, provide methods to enter details of medicine shops and medicine stores available in every hospital available in each district, provide a method to keep track of the type of medicines consumed in each district. Also to provide a threshold value for each category of medicine to predict any kind of epidemic disease for that district.

Concepts used in the Project

- Classes
- Objects
- Dynamic Memory Allocation
- String Manipulation
- File Handling
- Some Linux Libraries

CLASS DIAGRAMS

1. ADMIN Class:



Int checkid(char token[ ]) : This function make sure that when a new user registers a medical store , a unique id is given to him.

int Admin::check2(): It is used to check the if the name of district pre-exist in "soldist" file or not, .if not write in file

void changefile(string user,string y): It is used to edit the quantity of medicine in addmedicine() by creating a temporary file

void Admin::pepi(): It helps in predicting the epidemic by checking the maximum number of a particular medicine sold

changedist(string user,string ser2,int q): It checks the specific district for a medicine if it exists or not , if it exists then it add the quantity sold otherwise adds the medicine

void admindist(string user,string searc,int q): This function is used to add the sold medicine details to each district;s specific file

ADMIN

This class contain functions that make sure unique id and username are checked during login. It also determines the epidemic spread.

Int checkMedicine(char token[ ]): It makes sure that the name of a medicine is not repeated twice.

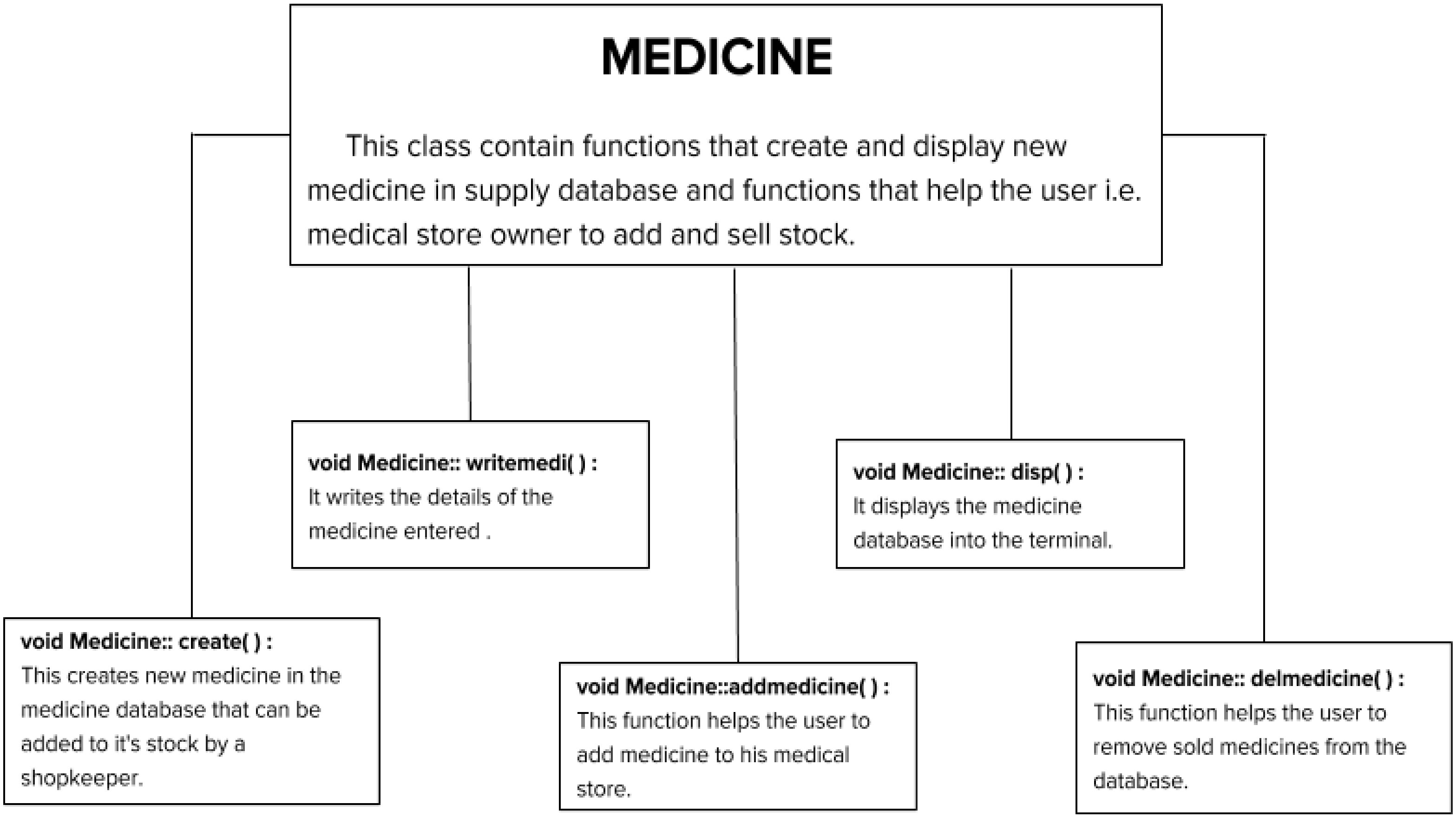
void epidemic(): It displays the name and quantity of medicine sold in each district and predicts epidermic based on largest selling medicine in the district

uni(char token[ ]): This function checks the "unique" keyword which helps in login.

int check(string s1, string s2): it takes two parameters string1 and string 2 ....string1 is the name of file ,it opens the file and return 1 if string is found in string 1 file

Int checkdistrict(char token[ ]): It ensures that same district is not repeated more than one time.

2. MEDICINE Class:



MEDICINE

This class contain functions that create and display new medicine in supply database and functions that help the user i.e. medical store owner to add and sell stock.

void Medicine:: writemedi( ) : It writes the details of the medicine entered .

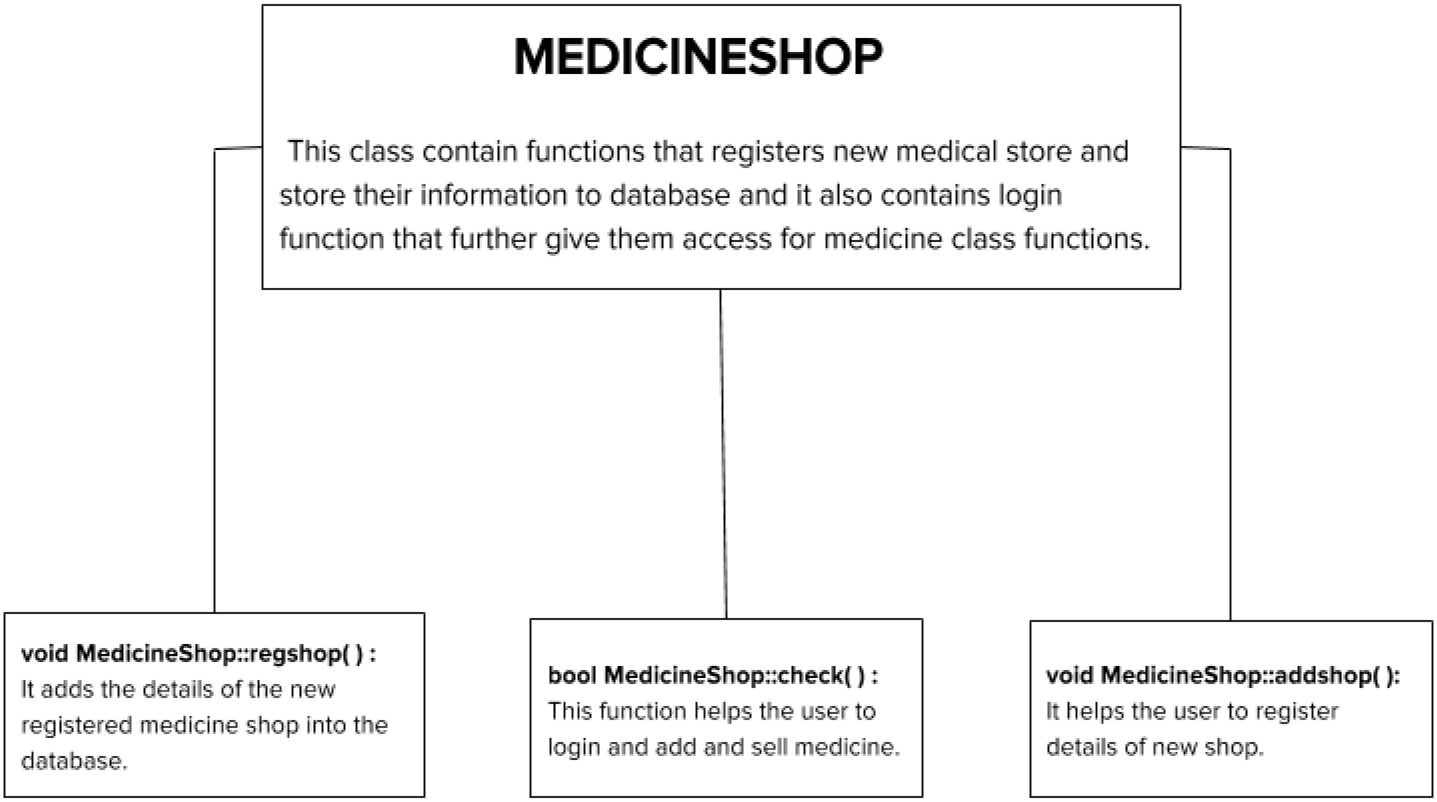
void Medicine:: disp( ) : It displays the medicine database into the terminal.

void Medicine:: create( ) : This creates new medicine in the medicine database that can be added to it's stock by a shopkeeper.

void Medicine::addmedicine( ) : This function helps the user to add medicine to his medical store.

void Medicine:: delmedicine( ) : This function helps the user to remove sold medicines from the database.

3. MEDICINESHOP Class:



MEDICINESHOP

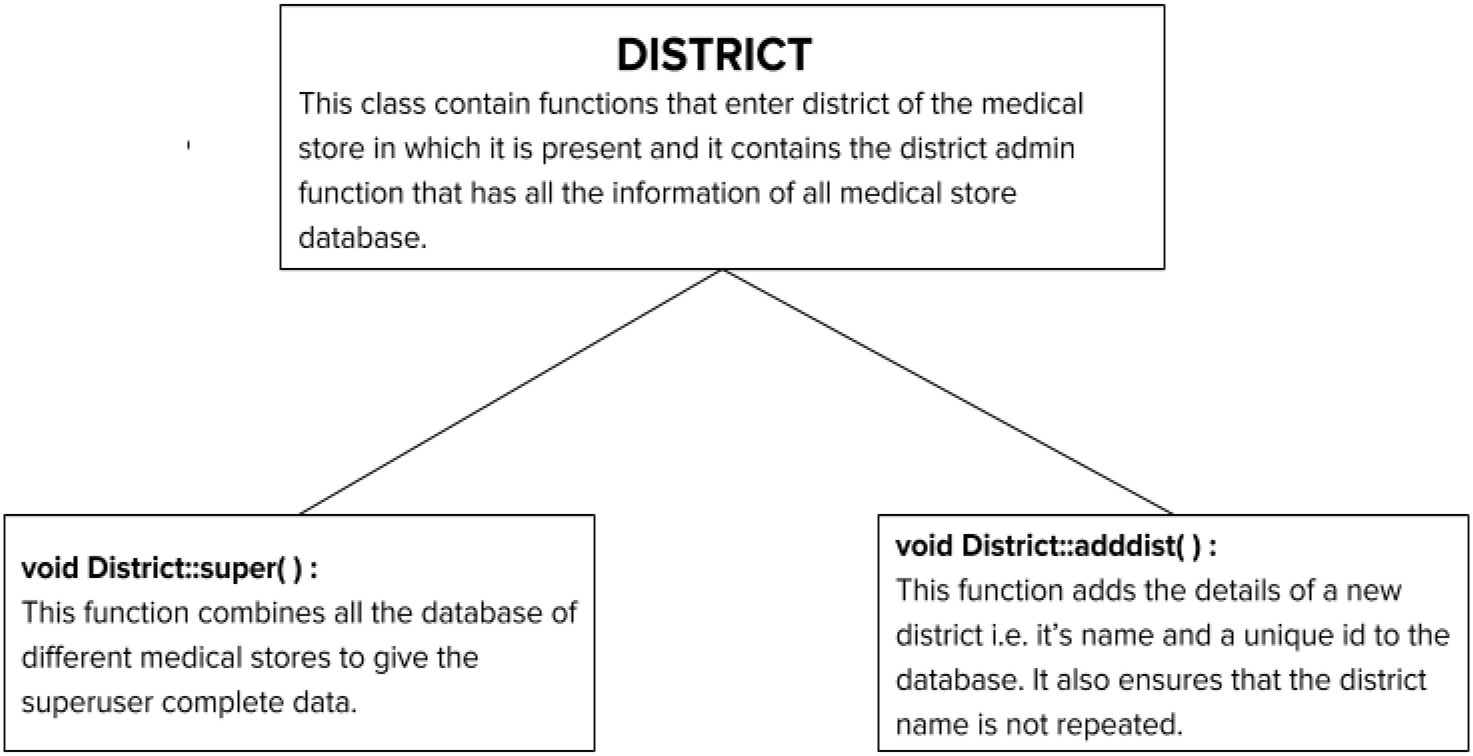
This class contain functions that registers new medical store and store their information to database and it also contains login function that further give them access for medicine class functions.

void MedicineShop::regshop( ) : It adds the details of the new registered medicine shop into the database.

bool MedicineShop::check( ) : This function helps the user to login and add and sell medicine.

void MedicineShop::addshop( ): It helps the user to register details of new shop.

4. DISTRICT Class:



DISTRICT

This class contain functions that enter district of the medical store in which it is present and it contains the district admin function that has all the information of all medical store database.

void District::super( ) :

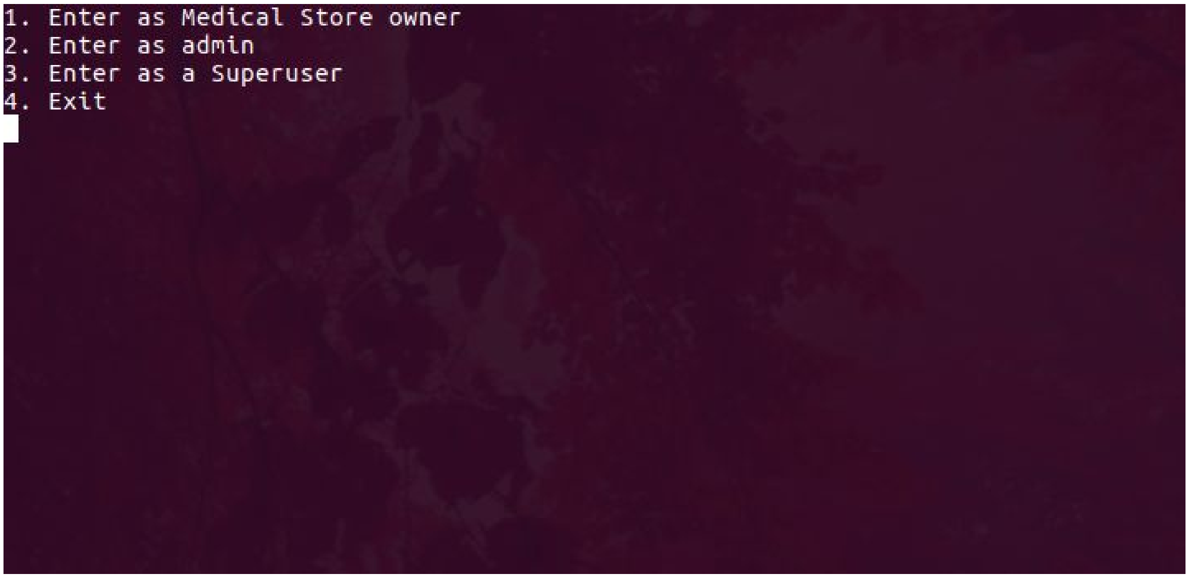
This function combines all the database of different medical stores to give the superuser complete data.

void District::adddist( ) :

This function adds the details of a new district i.e. it's name and a unique id to the database. It also ensures that the district name is not repeated.

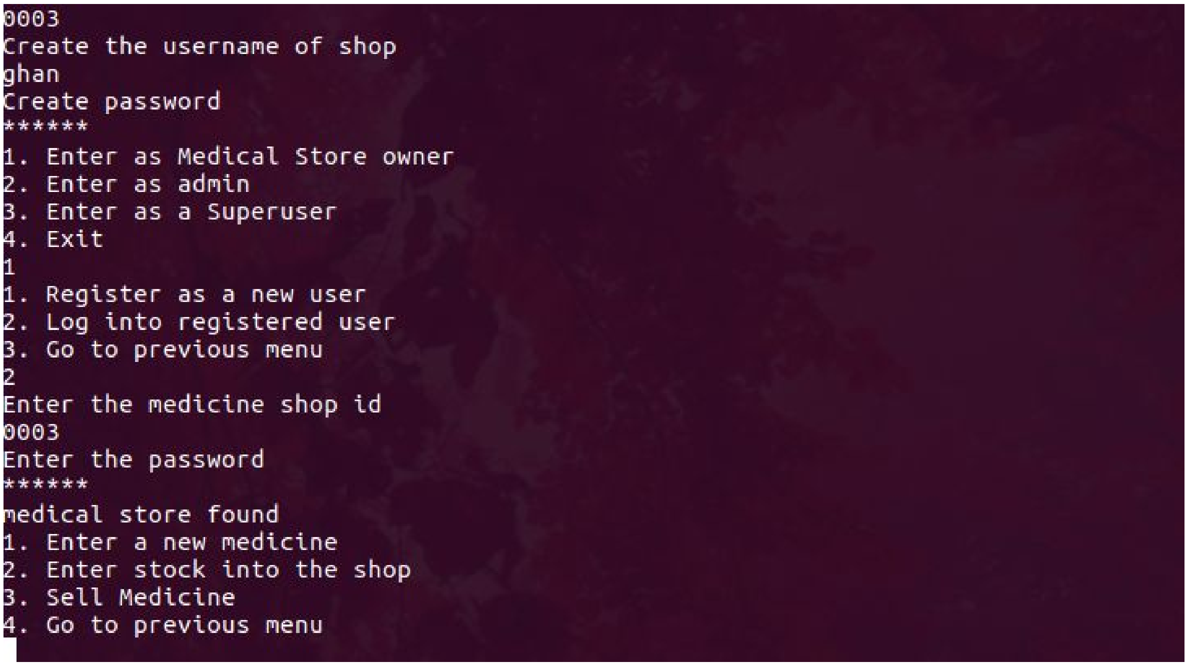
Snapshots of Interfaces

1. Menu:



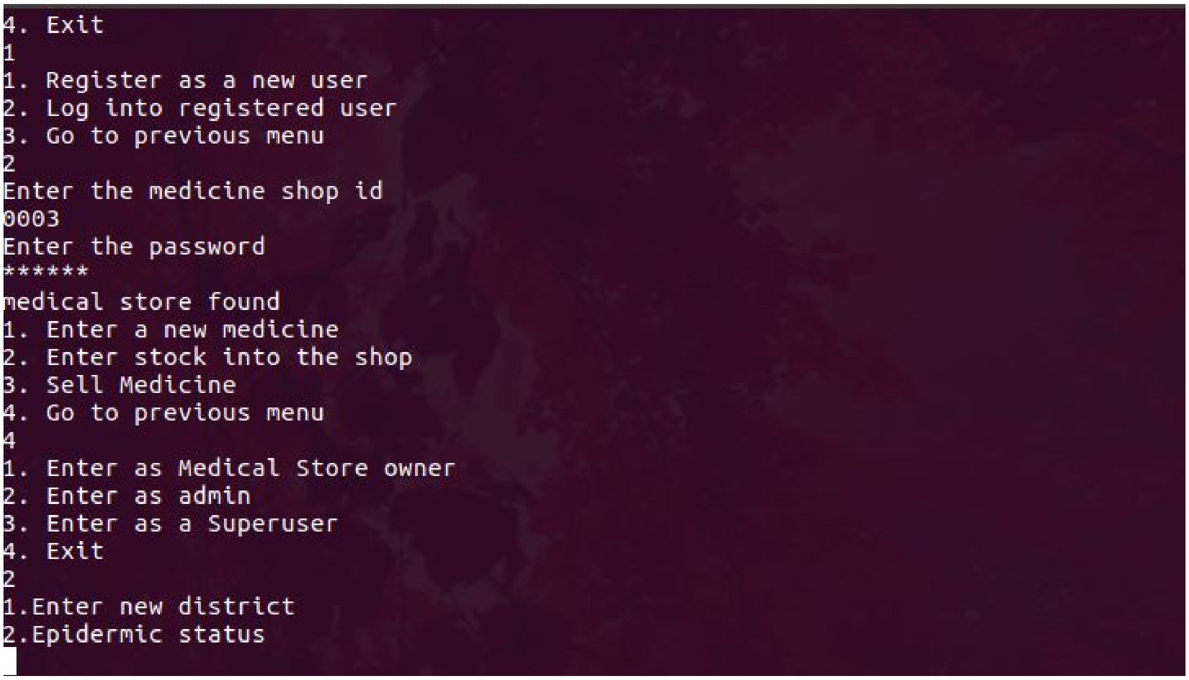
```
1. Enter as Medical Store owner
2. Enter as admin
3. Enter as a Superuser
4. Exit
```

2. Login User:



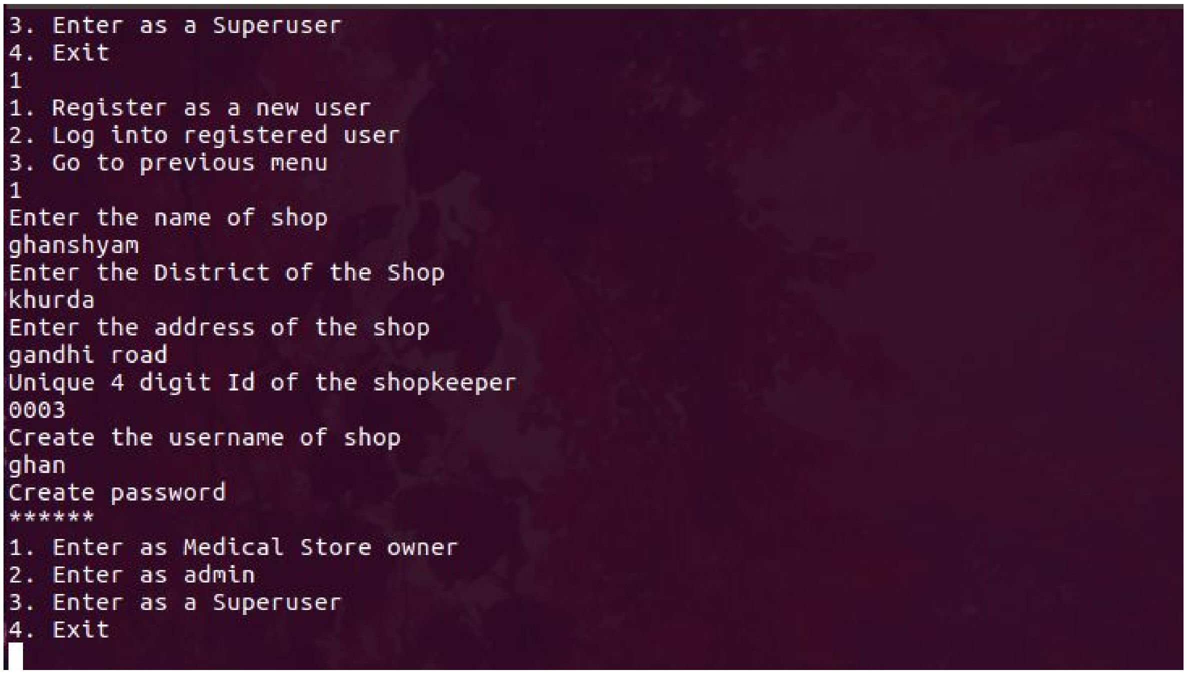
```
0003
Create the username of shop
ghan
Create password
******
1. Enter as Medical Store owner
2. Enter as admin
3. Enter as a Superuser
4. Exit
1
1. Register as a new user
2. Log into registered user
3. Go to previous menu
2
Enter the medicine shop id
0003
Enter the password
******
medical store found
1. Enter a new medicine
2. Enter stock into the shop
3. Sell Medicine
4. Go to previous menu
```

3. Admin:



```
4. Exit
1
1. Register as a new user
2. Log into registered user
3. Go to previous menu
2
Enter the medicine shop id
0003
Enter the password
******
medical store found
1. Enter a new medicine
2. Enter stock into the shop
3. Sell Medicine
4. Go to previous menu
4
1. Enter as Medical Store owner
2. Enter as admin
3. Enter as a Superuser
4. Exit
2
1.Enter new district
2.Epidermic status
```

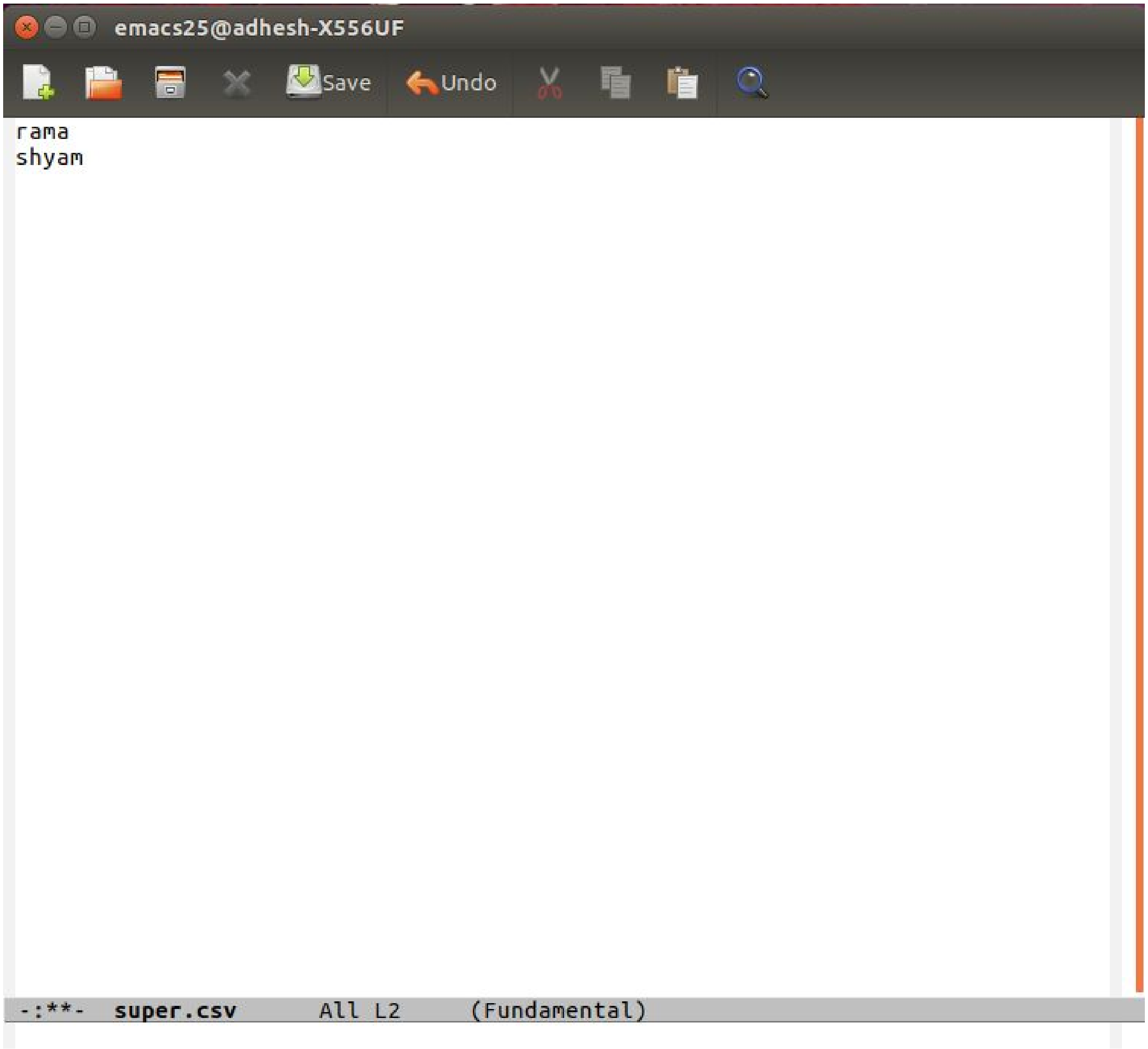
4. Register:



```
3. Enter as a Superuser
4. Exit
1
1. Register as a new user
2. Log into registered user
3. Go to previous menu
1
Enter the name of shop
ghanshyam
Enter the District of the Shop
khurda
Enter the address of the shop
gandhi road
Unique 4 digit Id of the shopkeeper
0003
Create the username of shop
ghan
Create password
******
1. Enter as Medical Store owner
2. Enter as admin
3. Enter as a Superuser
4. Exit
```

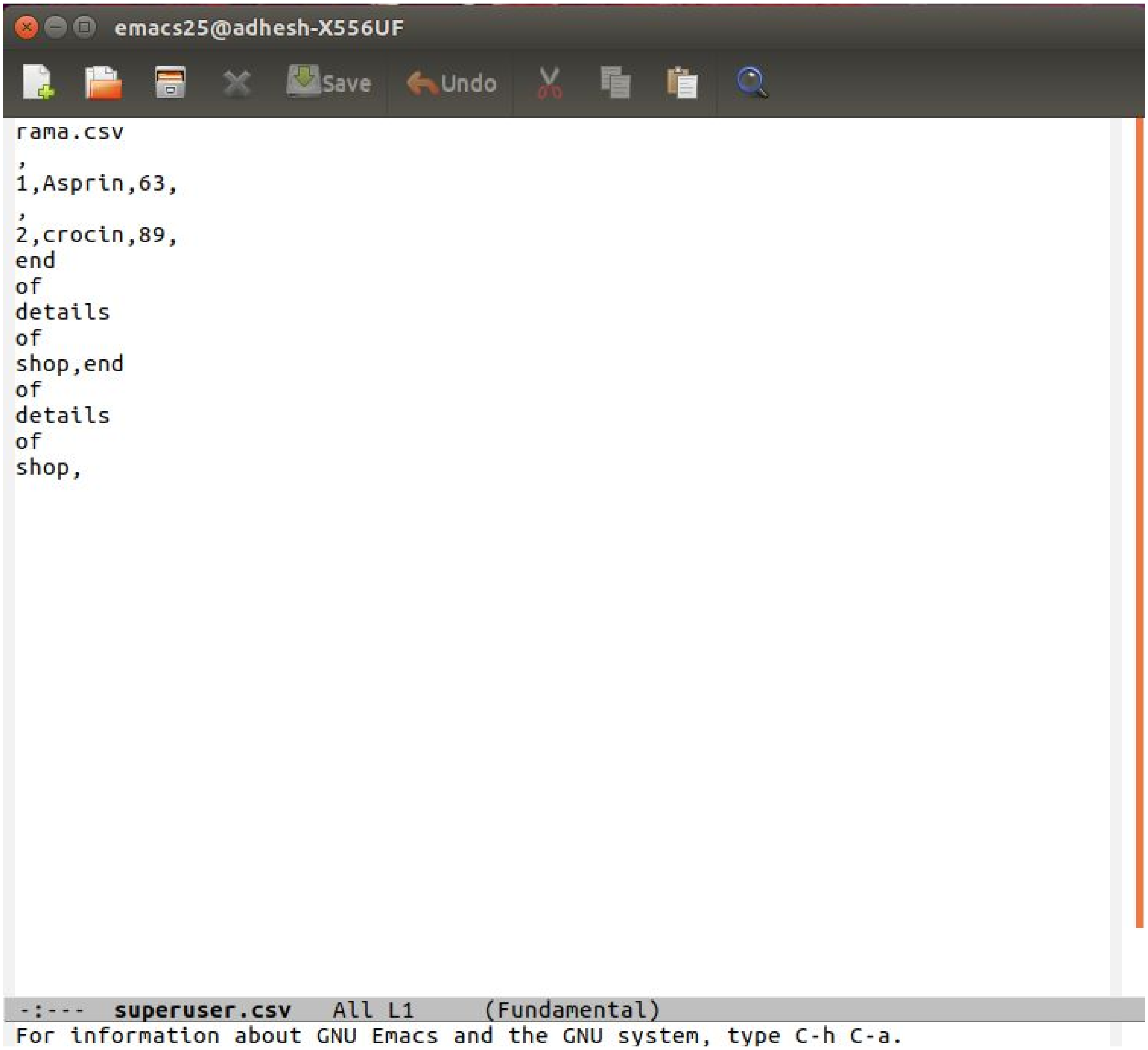
Snapshots of Backend Files

1. super.csv



```
rama
shyam
```

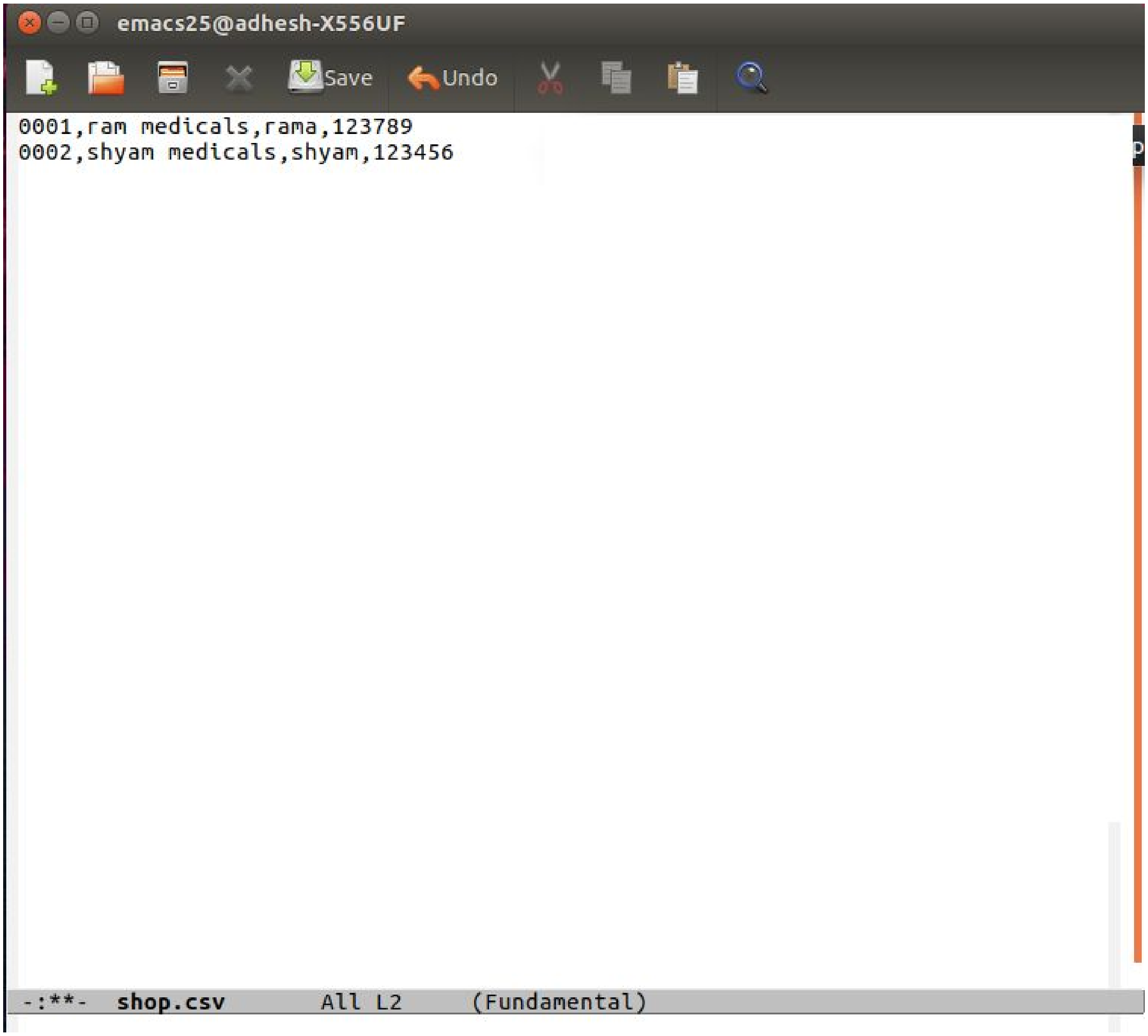
2. superuser.csv



```
rama.csv
,
1,Asprin,63,
,
2,crocin,89,
end
of
details
of
shop,end
of
details
of
shop,
```

```
-:---  superuser.csv   All L1     (Fundamental)
For information about GNU Emacs and the GNU system, type C-h C-a.
```

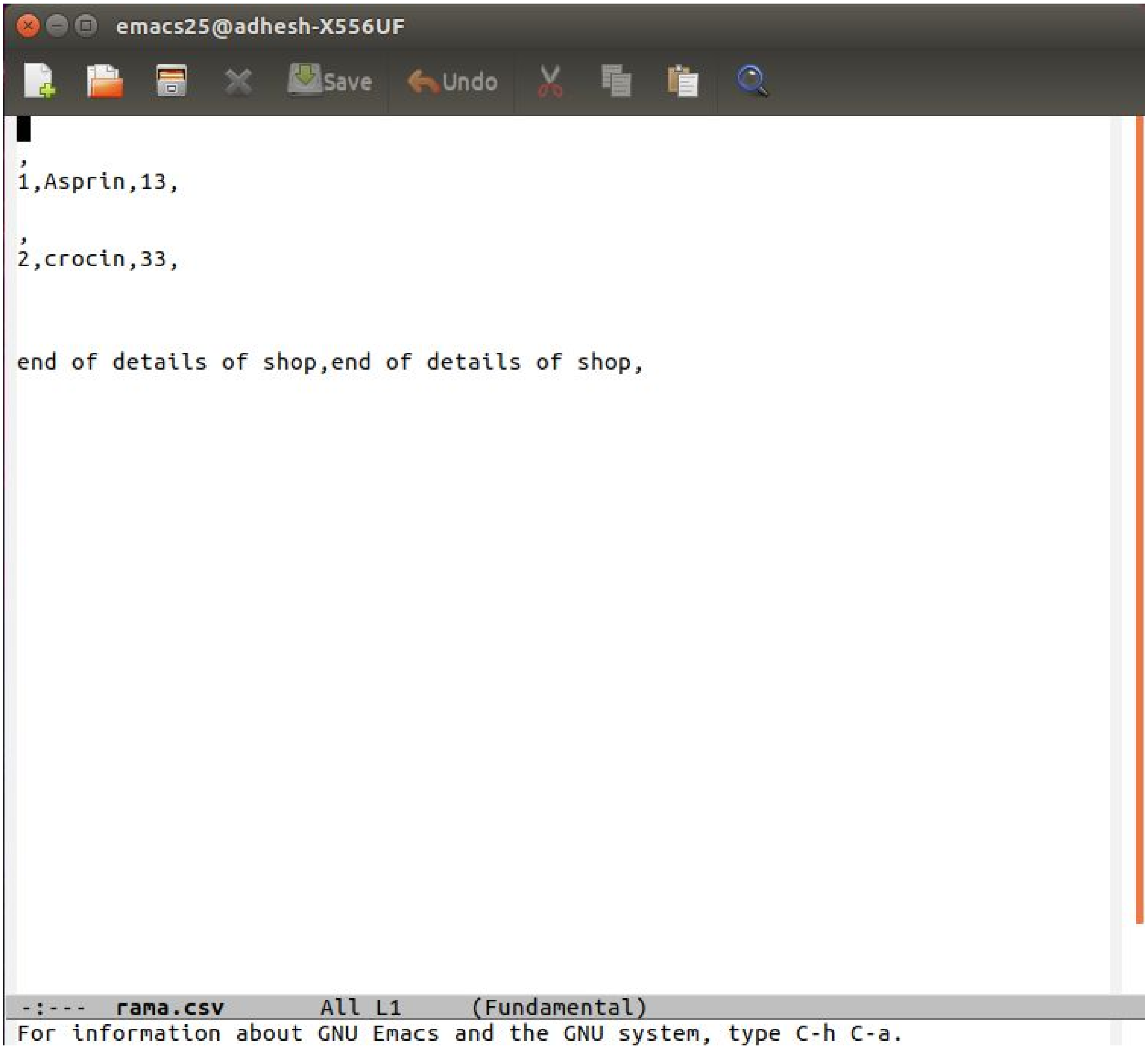
3. shop.csv



```
0001,ram medicals,rama,123789
0002,shyam medicals,shyam,123456
```

```
-:**-  shop.csv       All L2     (Fundamental)
```

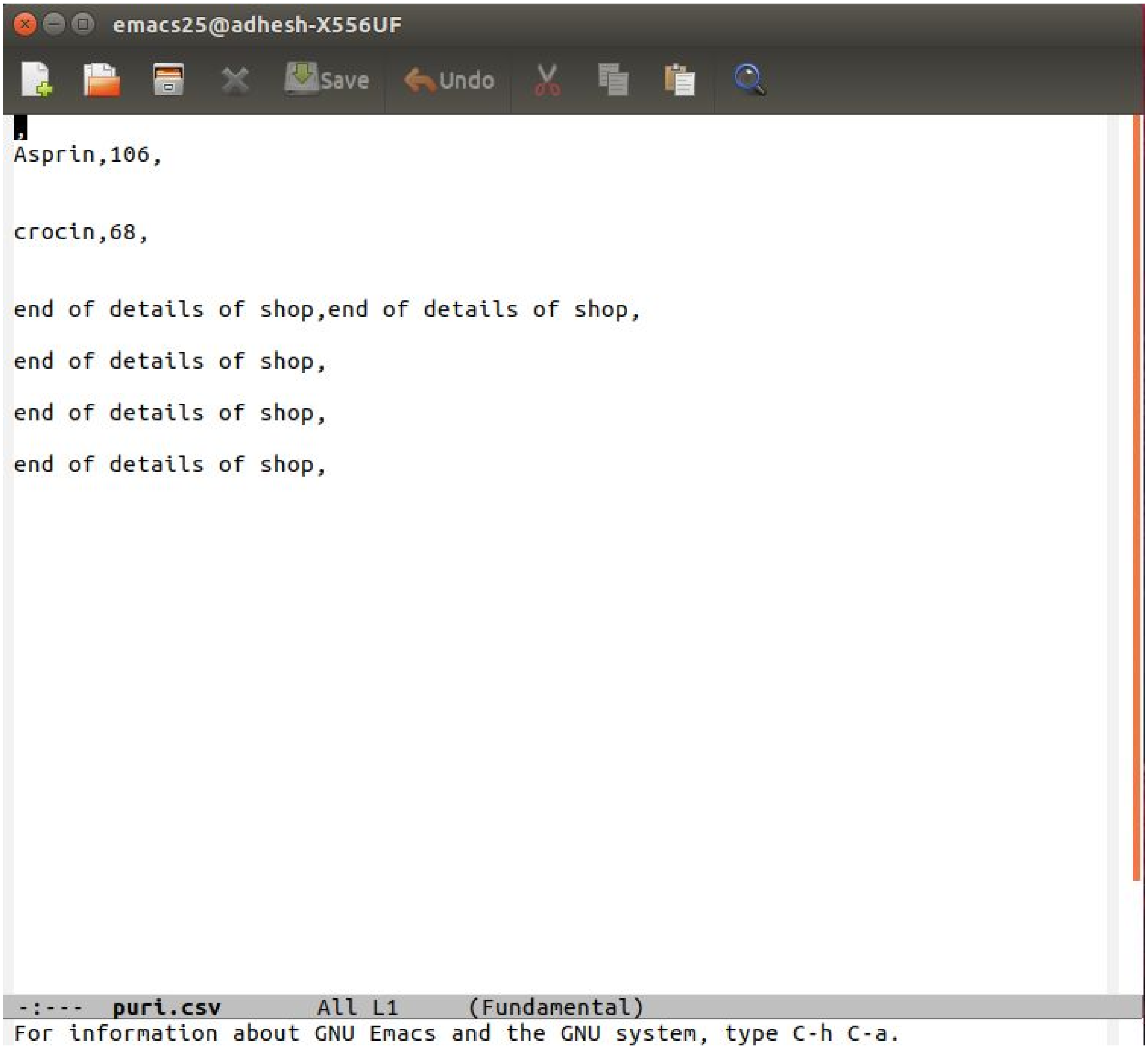
4. rama.csv:



```
,
1,Asprin,13,

,
2,crocin,33,



end of details of shop,end of details of shop,
```

```
-:---  rama.csv       All L1     (Fundamental)
For information about GNU Emacs and the GNU system, type C-h C-a.
```

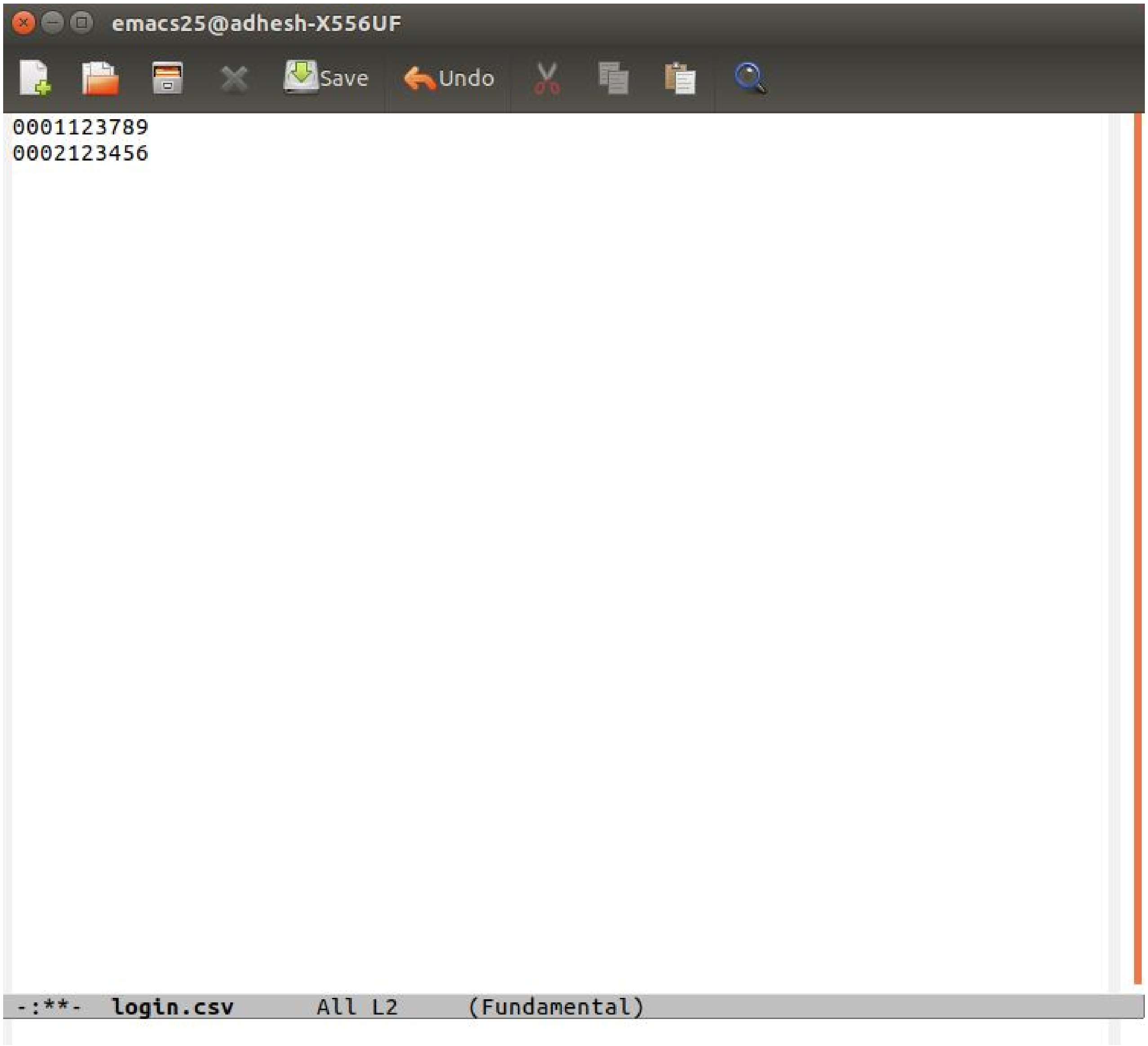
5. Puri.csv:



```
,
Asprin,106,


crocin,68,


end of details of shop,end of details of shop,

end of details of shop,

end of details of shop,

end of details of shop,
```

```
-:---  puri.csv       All L1      (Fundamental)
For information about GNU Emacs and the GNU system, type C-h C-a.
```

6. login.csv:



```
0001123789
0002123456
```

Source Code

```
#include<iostream>
#include<fstream>
#include<string>
#include<cstring>
#include<sstream>
#include<termios.h>
#include<unistd.h>

using namespace std;
int getch()
 {
    struct termios oldt, newt;
    int ch;
    tcgetattr(STDIN_FILENO, &oldt);
    newt = oldt;
    newt.c_lflag &= ~(ICANON | ECHO);
    tcsetattr(STDIN_FILENO, TCSANOW, &newt);
    ch = getchar();
    tcsetattr(STDIN_FILENO, TCSANOW, &oldt);
    return ch;
}
class Admin
{
    static int c;

public:
          void pepi()
               {
                string line1,line,line3,ar,dist;
                int l = 0;
                int pu;
                fstream fp,tp;
                tp.open("soldist.csv",ios::in);
                while(!tp.eof())
                {
                    l =0;
                    getline(tp,line1,',');
                    dist = line1;
                    line1+=".csv";
                    fp.open(line1,ios::in);
                     while(!fp.eof())
                    {
```

```
                  getline(fp,line,',');
                  stringstream geek(line);
                  pu = 0;
                  geek >> pu;
                  if(pu>0)
                   {
                     if(pu>l)
                       {
                        l = pu;
                        ar = line;
                       }
                   }
              }
              fp.close();

              if(l>0)
                {
                  ifstream stream;
                  stream.open(line1,ios::in);
                  while (getline(stream, line))
                       {
                         if (line.find(ar) != string::npos)
                           {
                            cout<<"District Name  -"<<dist<<endl;
                            cout<<"The maximum sold medicine in the district
is"<<endl<<line<<endl;
                            cout<<"The disease corresponding to this medicine is likely to
spread in the district"<<endl;
                           }
                       }

                  stream.close();

                }

               }
          }

     void epidermic()
      {
        fstream fp;
        char ch;
        string line,a,b,c;
        int choi;
        cout<<"1.View specific district's sold Medicines "<<endl<<"2.Predict epidermic
```

```
"<<endl;
           cin>>choi;
           switch(choi)
           {
              case 1:
               {
                   cout<<"Districts in which medicines are sold"<<endl;
                  fp.open("soldist.csv",ios::in);
                  while(!fp.eof())
                    {
                       getline(fp,line,',');
                       cout<<line;
                    }
            fp.close();
              do{
               cout<<"Enter the name of district to display its details"<<endl;
               cin>>a;
               int che = check2(a);
               a = (a+".csv");
               if(che==1)
               {
                   fp.open(a,ios::in);
                   cout<<"Medicine"<<" "<<"Quantity"<<endl;
                       while(!fp.eof())
                     {
                        getline(fp,line,',');
                       cout<<" "<<line;
                    }
                    fp.close();
                 }
              else
              {
              cout<<"NO such district exist"<<endl;
              }
          cout<<"want to see more details Y or N"<<endl;
          cin>>ch;


            }  while(ch=='y'||ch=='Y');
            break;
        }
          case 2: {
             Admin ob;
             ob.pepi();
             }
```

```
         break;
         }
  };
    int check2(string dist)
     {
       ifstream stream;
        stream.open("soldist.csv",ios::in);
        string line;
    while (getline(stream, line))
      {
        if (line.find(dist) != string::npos)
         {
           return 1;
         }
       }
       return 0;
    }

  void admindist(string user,string searc,int q)
    {
        user=user+".csv";
        int x,y,pu;
       fstream fp,tp,pt;
       string line,ser2,dist;
       if(c==0){
       fp.open(user,ios::app|ios::out);
       fp<<"end of details of shop"<<','<<endl;
       fp.close();
     c++;
     }
   Admin ob1;
   int check = ob1.check(searc,user);
    if(check==0)
   {
    ob1.changedist(user,searc,q);
   }
else
 {
 fp.open(user,ios::in|ios::out);
 tp.open("temp.csv",ios::app|ios::out);
 ser2 ="end of details of shop";
while(!fp.eof())
   {
       getline(fp,line,',');
       tp<<line<<',';
       if (line.find(searc) != string::npos||line.find(ser2) != string::npos)
    {


          break;

     }
   }
  getline(fp,line,',');
   int x = fp.tellg();
     {
      stringstream geek(line);
      pu = 0;
      geek >> pu;
      }
    q= (pu+q);
   ostringstream str1;
   str1 << q<<',';
   string geek = str1.str();
  tp<<geek;
  fp.seekp(x);
  tp<<endl;
  while(!fp.eof()){
  getline(fp,line);
   tp<<line<<endl;
  }
  fp.close();
  tp.close();
   char oldname[] ="temp.csv";
   char newname[100] ;
    strcpy(newname,user.c_str());
    int result = remove(newname);
     result = rename(oldname,newname);
  }
  }

int checkmedicine(char token[])

 {
  ifstream stream;
  stream.open("shop.csv",ios::in);
  string line;
  while (getline(stream, line))
   {

        if (line.find(token) != string::npos)
```

```
          {
             return 1;
          }
    }
         return 0;
 }
 int checkdistrict( char token[])
   {
      ifstream stream;
       stream.open("district.csv",ios::in);
       string line;
      while (getline(stream, line))
       {
        if (line.find(token) != string::npos)
         {
          return 1;
          }
       }

  return 0;
}
  int checkid(char token[])

  {
   ifstream stream;
   stream.open("shop.csv",ios::in);
   string line;
   while (getline(stream, line))
    {
         if (line.find(token) != string::npos)
          {
             return 1;
          }
    }
         return 0;
 }
 int uni(char token[])

 {
   ifstream stream;
   stream.open("login.csv",ios::in);
   string line;
    while (getline(stream, line))
      {
         if (line.find(token) != string::npos)
```

```
     {
      return 1;
     }
   }
    return 0;
 }
int check(string s1,string s2)
{
   ifstream stream;
 stream.open(s2.c_str(),ios::in);
 string line;
 while (getline(stream, line))
  {
    if (line.find(s1) != string::npos)
     {
      return 1;
     }
  }
    return 0;

}
void changedist(string user,string ser2,int q)
{
        fstream fp,tp;
        int count;
        string line,ser3;
        ser3 = "end of details of shop";
        fp.open(user,ios::in|ios::out);
        tp.open("u.csv",ios::app|ios::out);

      while(!fp.eof())
       {
         getline(fp,line,',');
         if (line.find(ser3) != string::npos)
         {
          break;
          }
       tp<<','<<line<<',';
      }

  tp<<endl;
  tp<<ser2<<','<<q<<','<<endl;
  tp<<line<<',';
```

```
   fp.close();
   tp.close();
    char oldname[] ="u.csv";
    char newname[100] ;
    strcpy(newname,user.c_str());
    int result = remove(newname);
   result = rename(oldname,newname);
 }


void changefile(string user,string y )
  {
   fstream fp,tp;
   int count;
   string name,ser2,line;
    ser2 = "end of details of shop";
    int q;
   fp.open(user,ios::in|ios::out);
   tp.open("u.csv",ios::app|ios::out);

  while(!fp.eof())
  {
     getline(fp,line,',');
        if (line.find(ser2) != string::npos)
       {
          count = 1;
           break;
       }
   tp<<line<<',';
  }
    tp<<endl;
     cout<<"enter the qanti"<<endl;
     cin>>q;
   cout<<"enter id"<<endl;
   int id;
   cin>>id;
   tp<<id<<','<<y<<','<<q<<','<<endl;
   tp<<line<<',';
   fp.close();
   tp.close();
    char oldname[] ="u.csv";
   char newname[100] ;
   strcpy(newname,user.c_str());
   int result = remove(newname);
   result = rename(oldname,newname);
```

```
}
};
int Admin::c;

class Medicine
  {
   private:
       string name,exp,number,a;
       int d,m,y;
   public:
       fstream fp;
       void create()
        {
           cout<<"Enter the Name of the Medicine "<<endl;
           getline(cin,a);
           getline(cin,name);
           cout<<"Enter the Medicine Id"<<endl;
           getline(cin,number);
           cout<<"Enter the expiry date of the medicine in the form dd/mm/yyyy"<<endl;
           cin>>d;
           while(1)
             {
               if ( cin.get() != '/' )
                  {
                    cout << "expected /\n";
                  }
               else
                  break;
             }
           cin >> m;
           while(1)
           {
            if ( cin.get() != '/' )
              {
                cout << "expected /\n";
              }
            else
              break;
           }
            cin >> y;
            exp=to_string(d)+"/"+to_string(m)+"/"+to_string(y);
        }

       void writemedi()
        {
```

```
   fp.open("MEDICINE.csv",ios::out|ios::app);
      {
        fp<<number<<','<<name<<','<<exp<<','<<endl;
         }
   fp.close();
 }
void disp()
 {
   char s[50];
    fp.open("MEDICINE.csv",ios::in);
     {
      while (true)
       {
        fp >> s;
        if( fp.eof() )
            break;
        cout<<s<<endl;
       }
     }
    fp.close();
 }
void addmedicine()
 {
   string user;
   cout<<"Enter the username"<<endl;
   getline(cin,a);
   getline(cin,user);
   user=user+".csv";
   int x,y,q,pu;
   fstream fp,tp,pt;
   string line,searc,ser2,dist;
   fp.open(user,ios::app|ios::out);
   fp<<"End of details of shop"<<','<<endl;
   fp.close();
   cout<<"Enter the name of medicine"<<endl;
   cin>>searc;
   Admin ob1;
   int check = ob1.check(searc,user);
   if(check==0)
    {
      ob1.changefile(user,searc);
    }
   else
    {
      fp.open(user,ios::in|ios::out);
```

```
       tp.open("SHOP.csv",ios::app|ios::out);
       ser2 ="end of details of shop";
       while(!fp.eof())
         {
          getline(fp,line,',');
          tp<<line<<',';
          if (line.find(searc) != string::npos||line.find(ser2) != string::npos)
            break;
         }
       getline(fp,line,',');
       x = fp.tellg();
       cout<<line<<endl;

    {
      stringstream geek(line);
       pu = 0;
      geek >> pu;
      cout << "Value of x : " << pu;
     }
      cout<<"Enter the quantity of Medicine to be added"<<endl;
      cin>>q;
      y = pu+q;
      ostringstream str1;
      str1 << y<<',';
      string geek = str1.str();
      tp<<geek;
      fp.seekp(x);
      tp<<endl;
      while(!fp.eof())
       {
        getline(fp,line);
        tp<<line<<endl;
       }
      fp.close();
      tp.close();
      char oldname[] ="SHOP.csv";
      char newname[100] ;
      strcpy(newname,user.c_str());
      int result = remove(newname);
      result = rename(oldname,newname);
   }
}

   void delmedicine()
   {
```

```
  string user,dist;
  cout<<"enter the name of district in which medicine shop is located"<<endl;
  cin>>dist;
   cout<<"Enter the username"<<endl;
 getline(cin,a);
 getline(cin,user);
 user=user+".csv";
  int x,y,q,pu;
   fstream fp,tp,pt,qt;
   string line,searc,ser2;
  fp.open(user,ios::in|ios::out);
  tp.open("SHOP.csv",ios::app|ios::out);
  cout<<"enter the name of medicine"<<endl;
 cin>>searc;
 Admin ob1;
 int check = ob1.check(searc,user);
if(check==0)
   {
    cout<<"No medicine available"<<endl;
   }
else{
   ser2 ="end of details of shop";
   while(!fp.eof())
     {
      getline(fp,line,',');
      tp<<line<<',';
      if (line.find(searc) != string::npos||line.find(ser2) != string::npos)
         {
             break;
          }
       }
getline(fp,line,',');
 x = fp.tellg();
 cout<<line<<endl;
{stringstream geek(line);
 pu = 0;
 geek >> pu;
  cout << "Value of x : " << pu<<endl;
  }
 while(1)
  {
   cout<<"Enter the quantity of Medicine to Sell"<<endl;
   cin>>q;
   if(q<pu)
    {
```

```
               break;
              }
             if(q>pu)
               {
                cout<<"That much quantity not available"<<endl;
               }
             }
             ob1.admindist(dist,searc,q);
             qt.open("soldist.csv",ios::app|ios::out);
             string d;int c;
              c =ob1.check2(dist);
             if(c==0)
             {
               qt<<dist<<','<<endl;
              }
              qt.close();
              y = pu-q;
              ostringstream str1;
               str1 << y<<',';
              string geek = str1.str();
             tp<<geek;
             fp.seekp(x);
             tp<<endl;
             while(!fp.eof())
             {
               getline(fp,line);
               tp<<line<<endl;
             }
             fp.close();
              tp.close();
             char oldname[] ="SHOP.csv";
             char newname[100] ;
             strcpy(newname,user.c_str());
            int result = remove(newname);
             result = rename(oldname,newname);
          }
        }
};
class District
{
private:
    string dist_name;
    char num[10];
public:
    void adddist()
```

```
 {

   fstream fp;
   string ar;
   char ch;
   ch = 'y';
   do{
   int i = 0;
   string unil;
   cout<<"Enter the name of the district"<<endl;
   getline(cin,ar);
   getline(cin,dist_name);
   cout<<"Enter the unique number of district of format D-Num"<<endl;
   while(i<6)
   {
     num[i]=getch();
     cout<<"&";
     i++;
   }
   cout<<endl;
   unil.append(num);
   Admin ob;
   char ar[50];
   strcpy(ar,dist_name.c_str());
   int che= ob.checkdistrict(ar);
   if(che==1)
   {
       cout<<"District with same name exist"<<endl;
    }
   else
   {
     fp.open("district.csv",ios::out|ios::app);
                 {
                            fp<<unil<<','<<dist_name<<','<<endl;
       }
  cout<<"Want to enter more districts- Y or N"<<endl;
   cin>>ch;
   fp.close();
   }
 }while(ch=='y'||ch=='Y');


}
 void super()
 {
        fstream fp,fq,fw;
        string s;
        char e[100];
        fp.open("super.csv",ios::in);
        fq.open("superuser.csv",ios::out);
        {
          while (true)
          {
         fp >> s;
         if( fp.eof() )
         break;
         s=s+".csv";
         fw.open(s.c_str(),ios::in);
         fq<<s<<endl;
             while (true)
              {
                fw>>e;
                if( fw.eof() )
                 break;
                fq<<e<<endl;
              }
              fq<<endl;
                fw.close();
        }
      }
        fp.close();
        fq.close();

      }
};

class MedicineShop
{
private:
    string name,ar,password,add,unique,dist;
public:
    Admin ad;
    fstream fp,fq,fj;
    char Id[300],username[300];
    char pass[20],a;
    int u,i=0;
    void addshop()
        {
          getline(cin,ar);
          cout<<"Enter the Name of the Shop"<<endl;
```

```
 getline(cin,name);
 cout<<"Enter the District of the Shop"<<endl;
 getline(cin,dist);
 cout<<"Enter the Address of the shop"<<endl;
 getline(cin,add);

 while(1)
   {
     cout<<"Unique 4 digit Id of the shopkeeper"<<endl;
     cin>>Id;
     int i = ad.checkid(Id);
     if(i==1)
      {
        cout<<"Please Re-enter Id"<<endl;
      }
     else
        break;
   }
 while(1)
   {
     cout<<"Create the Username of the Shop"<<endl;
     cin>>username;
     int i = ad.checkid(username);
     if(i==1)
      {
        cout<<"Please Re-enter Id"<<endl;
      }
     else
       break;
   }

 cout<<"Create Password"<<endl;
 a=getch();
 while(i<6)
   {
    pass[i]=getch();
    cout<<'*';
    i++;
   }
 cout<<endl;
 password.append(pass);
 unique=Id+password;

}
```

```
void regshop()
  {
    fp.open("shop.csv",ios::out|ios::app);
    fj.open("login.csv",ios::out|ios::app);
    fq.open("super.csv",ios::out|ios::app);
        {
            fp<<Id<<','<<name<<','<<username<<','<<password<<endl;
            fj<<unique<<endl;
            fq<<username<<endl;
        }
    fp.close();
    fq.close();
    fj.close();
  }

bool check()
  {
    string s,s2,s3,s4;
    char p[10];
    cout<<"Enter the Medicine Shop Id"<<endl;
    getline(cin,s4);
    getline(cin,s);
    cout<<"Enter the Password"<<endl;
    while(i<6)
      {
        p[i]=getch();
        cout<<'*';
        i++;
      }
    cout<<endl;
    s2.append(p);
    s3=s+s2;
    char* y=new char[s3.length() + 1];
    strcpy(y, s3.c_str());
    if(ad.uni(y))
      {
        cout<<"Medical store found"<<endl;
        return 1;
      }
    else
      {
        cout<<"Medical Store not found!"<<endl;
        return 0;
      }
    delete[] y;
  }
};

int main()
{
 int l,s=1,o,i,r=1;
 while(s!=0)
 {
  cout<<"1. Enter as Medical Store Owner"<<endl<<"2. Enter as Admin "<<endl<<"3.
Enter as a Superuser"<<endl<<"4. Exit"<<endl;
  cin>>l;
   switch(l)
   {
    case 1:
     {
      cout<<"1. Register as a New User"<<endl<<"2. Log-in as a Registered User
"<<endl<<"3. Go to the Previous Menu"<<endl;
      cin>>o;
      switch(o)
      {
       case 1:
       {
        MedicineShop *m=new MedicineShop();
        m->addshop();
        m->regshop();
        delete m;
        break;
       }
       case 2:
      {
       MedicineShop *t=new MedicineShop();
       if(t->check())
       {
       while(r!=0)
        {
       cout<<"1. Enter a new medicine "<<endl<<"2. Enter stock into the
shop"<<endl<<"3. Sell Medicine"<<endl<<"4. Go to previous menu"<<endl;
       cin>>i;
       switch(i)
       {
        case 1:
        {
         Medicine *q=new Medicine();
         q->create();
         q->writemedi();
```

```
            delete q;
            break;
          }
          case 2:
          {
            Medicine *p=new Medicine();
            p->disp();
            p->addmedicine();
            delete p;
            break;
          }
          case 3:
          {
            Medicine *r = new Medicine();
            r->delmedicine();
            delete r;
            break;
          }
          case 4:
          {
            r=0;
            break;
          }
          default:
          {
            cout<<"Invalid input try again "<<endl;
          }
         }
      }
    }
     delete t;
     break;
    }
  }
  break;
}
case 2:
{
  cout<<"1.Enter a New District"<<endl<<"2.Epidemic Status"<<endl;
  int choice;
  cin>>choice;
  switch(choice)
  {
         case 1:
         {
```

```
                    District *dp=new District();
                    dp->adddist();
                    delete dp;
                    break;
                }
                case 2:
                {
                    Admin *d=new Admin();
                    d->epidermic();
                    delete d;
                    break;

                }
        }

        break;
      }

      case 3:
          {
            District *d=new District();
            d->super();
            delete d;
            break;
          }
      case 4:
      {
        s=0;
        break;
      }
     }
    }
  return 0;
}
```

References

The project has been completed under the supervision and guidance of our Object Oriented Programming faculty Mr. Muktikanta Sahu.

The following have further helped us in the completion of the project:

Books:

- Object Oriented Programming with C++, by - E. Balagurusamy

Websites:

- https://stackoverflow.com
- https://askubuntu.com
- https://ubuntuforums.org
- http://www.cplusplus.com