**Protection of information based on sensitivity and privilege levels**

**CS3052 - Computer Security**

**190145G – W.A.M. Dilshan**

# The program codes

import csv

import hashlib

# mapping of privileges to privilege levels

read\_privileges = {

    'patient' : ['record\_id','user\_name','personal\_details','sickeness\_details','drug\_prescriptions','lab\_test\_prescriptions'],

    'lab staff' : ['record\_id','user\_name','personal\_details','lab\_test\_prescriptions'],

    'pharmacy staff' : ['record\_id','user\_name','personal\_details','drug\_prescriptions'],

    'nurse' : ['record\_id','user\_name','personal\_details','sickeness\_details','drug\_prescriptions','lab\_test\_prescriptions'],

    'doctor' : ['record\_id','user\_name','personal\_details','sickeness\_details','drug\_prescriptions','lab\_test\_prescriptions']

}

allowed\_actions = {

    'patient' : ['view my records'],

    'lab staff' : ['view all records', 'view all records by user name', 'view record by id'],

    'pharmacy staff' : ['view all records', 'view all records by user name', 'view record by id'],

    'nurse' : ['view all records', 'view all records by user name', 'view record by id'],

    'doctor' : ['view all records', 'view all records by user name', 'view record by id', 'add new record']

}

# hashing the given string using md5

def hash(str\_to\_hash):

    return hashlib.md5(str\_to\_hash.encode()).hexdigest()

# reading the config csv and store the data in the 2d array

def read\_csv(filename):

    data= []

    with open(filename,"r") as file:

        csvreader = csv.reader(file,delimiter=",")

        for row in csvreader:

            if row:

                data.append(row)

    return data

# writing a one row data to csv

def write\_csv(filename,data):

    with open(filename,"a") as file:

        csvwritter = csv.writer(file)

        csvwritter.writerow(data)

# validating if a given user exists already

def validate\_user\_exists(user\_details,username):

    for user\_data in user\_details:

        if user\_data[0] == username:

            return True,user\_data

    return False,[]

# function to handle login

def login():

    username = input("Enter your username: ")

    is\_username\_valid, user\_data = validate\_user\_exists(user\_details,username)

    if not is\_username\_valid:

        while True:

            choice = input("Entered username is invalid!\nPlease select a choice to continue: \n1 - Re-enter the login details\n2 - Go to main menu\n")

            if choice == "1":

                return login()

            elif choice == "2":

                return False,[]

            else:

                print("Incorrect input. Please try again!\n")

    password = input("Enter your password: ")

    if not user\_data[1] == hash(password):

        while True:

            choice = input("Entered username is invalid!\nPlease select a choice to continue: \n1 - Re-enter the login details\n2 - Go to main menu\n")

            if choice == "1":

                login()

            elif choice == "2":

                return False,[]

            else:

                print("Incorrect input. Please try again!\n")

    print("Login successfull!")

    return True,user\_data

# function to handle registeration

def register():

    global user\_details

    username = input("Enter a username: ")

    is\_user\_valid,\_ =  validate\_user\_exists(user\_details,username)

    if is\_user\_valid:

        print("This username already exists. Please try a different one.")

        return register()

    password = input("Enter a password: ")

    hashed\_password = hash(password)

    valid\_user\_type = False

    while not valid\_user\_type:

        user\_type\_choice = input("Enter the user type:\n1 - Patient\n2 - Hospital staff\n")

        valid\_user\_type = True

        if user\_type\_choice == "1":

            user\_type = "patient"

        elif user\_type\_choice == "2":

            user\_type = "hospital staff"

        else:

            print("Invalid input! Please try again.")

    if user\_type == "patient":

        privilege\_level = "patient"

    else:

        valid\_privilege\_level = False

        while not valid\_privilege\_level:

            privilege\_level\_choice = input("Enter your privilege level:\n1 - Lab staff\n2 - Pharmacy staff\n3 - Nurse\n4 - Doctor\n")

            valid\_privilege\_level = True

            if privilege\_level\_choice == "1":

                privilege\_level = "lab staff"

            elif privilege\_level\_choice == "2":

                privilege\_level = "pharmacy staff"

            elif privilege\_level\_choice == "3":

                privilege\_level = "nurse"

            elif privilege\_level\_choice == "4":

                privilege\_level = "doctor"

            else:

                print("Invalid input! Please try again.")

    user\_data = [username,hashed\_password,user\_type,privilege\_level]

    user\_details.append(user\_data)

    write\_csv("config.csv",user\_data)

    print("Registration successfull!")

    return True

# adding a new record

def add\_record():

    record\_id = str(len(record\_details)+1)

    username = input("Enter the patient username: ")

    is\_user\_valid,\_ =  validate\_user\_exists(user\_details,username)

    if not is\_user\_valid:

        print("This patient is not exist! please re-enter")

        return add\_record()

    patient\_details = input("Enter the patient details: ")

    sickness\_details = input("Enter the sickness details: ")

    drug\_description = input("Enter the drug description: ")

    lab\_test\_prescriptions = input("Enter the lab test prescriptions: ")

    record = [record\_id,username,patient\_details,sickness\_details,drug\_description,lab\_test\_prescriptions]

    write\_csv("record.csv",record)

    record\_details.append(record)

# getting records using record id

def get\_record\_by\_id(id):

    return record\_details[id-1]

# getting records using patient username

def get\_records\_by\_username(username):

    output = []

    for record in record\_details:

        if record[1] == username:

            output.append(record)

    return output

# helper function to show records to filter the fields in the records.

def filter\_record(records,user\_privilege):

    if user\_privilege == "lab staff":

        for record in records:

            record.pop(4)

    elif user\_privilege == "pharmacy staff":

        for record in records:

            record.pop(3)

    return records

# function to print the records with the filtering.

def show\_records(records,user\_privilege):

    print("Here the medical records!")

    filtered\_records = filter\_record(records,user\_privilege)

    for record in filtered\_records:

        print(" ".join(record))

    return

# generating action menu for given actions.

def generate\_menu(actions):

    menu ="select a choice from below options.\n"

    i = 0

    for action in actions:

        i += 1

        menu += str(i)+"- "+action+"\n"

    return menu

# this will run given command with username and user privilege. This where user's actions running after login and they selects a action.

def do\_action(action,username,user\_privilege):

    if action == "view my records":

        records = get\_records\_by\_username(username)

        show\_records(records,user\_privilege)

    elif action == "view all records":

        show\_records(record\_details,user\_privilege)

    elif action == "view all records by user name":

        req\_username = input("Enter the username of the patient: ")

        records = get\_records\_by\_username(req\_username)

        show\_records(records,user\_privilege)

    elif action == "view record by id":

        id = int(input("Enter the record ID: "))

        record = get\_record\_by\_id(id)

        show\_records([record],user\_privilege)

    elif action == "add new record":

        add\_record()

# loading neccessary data

user\_details = read\_csv("config.csv")

record\_details = read\_csv("record.csv")

# Here the main function

def main():

    is\_validate\_option = False

    user\_data = []

    while(not (is\_validate\_option and user\_data )):

        option = input("Select a relevent number for login or register\n1 - Login\n2 - Register\n")

        if option == "1":

            is\_validate\_option,user\_data=login()

        elif option == "2":

            is\_validate\_option= register()

        else:

            print("Invalid input. Please enter again choice 1 or 2.\n")

    username,user\_type,user\_privilege = user\_data[0],user\_data[2],user\_data[3]

    actions = allowed\_actions[user\_privilege]

    menu = generate\_menu(actions)

    while True:

        print(menu)

        choice = int(input("Enter your choice: "))

        if 1<=choice<=len(actions):

            do\_action(actions[choice-1],username,user\_privilege)

        else:

            print("Invalid input")

# running the main function.

main()

# A description of how you decided the access to data records based on sensitivity of data

There are three sensitivity levels in the program data set

1. Low sensitivity
2. Medium sensitivity
3. High sensitivity

For each sensitivity level has list of data fields such as

low= ['record\_id','user\_name','personal\_details'],

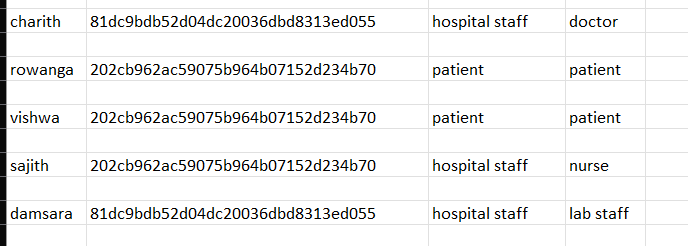
medium=['drug\_prescriptions','lab\_test\_prescriptions'],

high= ['sickeness\_details']

After the user login then program will generate the relevant action menu using the user privilege and then user will be able to select any option from the menu. Filtering happening according to the level of sensitivity. Write record only set to doctor.

# Annexes

Config file



Records file

Table

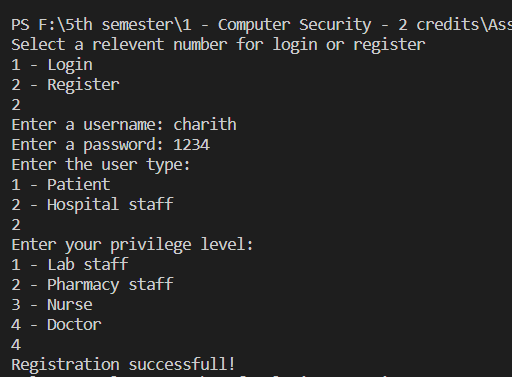
Description automatically generated

Login as a patient

Text

Description automatically generated

Register as a doctor

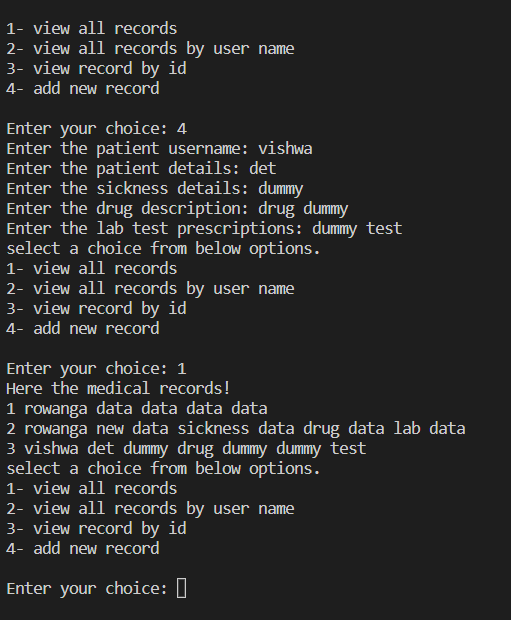


Register as a patient

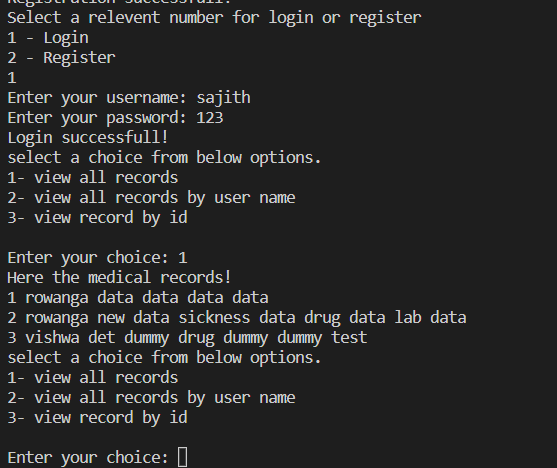
Text

Description automatically generated

Adding a new record and view all records as a doctor



View all records as a nurse



View all records as a lab staff

