Computer Science Project

-K Aditya (16)

CERTIFICATE:

This is to certify that K Aditya, Roll No 16 of class 12-D has successfully completed their project work on the topic “Persistent Database” for class 12 practical examination of CBSE in year 2020-2021.

ACKNOWLEDGEMENT:

In the accomplishment of this project, many people have best aimed upon me their blessings. I want to thank all the people who have helped me with this project.

I would like to thank my Computer Science teacher Mr. Amit Dua whose valuable guidance has been the one that helped me make this project.

I also extend my heartful thanks to my parents and friends who have inspired me to make the best out of this project.

CODE:

```
from pymysql import TIMESTAMP, cursors
import pymysql as sql
import os
import time

while True:  # Connecting to the databases
    try:
        # Password of the sql server
        passcode = input('Enter the password of your sql server: ')
        # Connecting to the user databases
        user_db = sql.connect(host='localhost', user='root',
                              passwd=passcode, db='users')
        # Creating a user cursor to handle all queries in the user database
        user_cursor = user_db.cursor()
        try:  # Creating the table if it doesn't exist
            user_cursor.execute("""create table users(
                                   username char(20) primary key,
                                   password char(20)
                               );""")
        except:
            pass
        try:
            user_cursor.execute("""create table messaginggroups(
                                   groupname char(20) primary key,
                                   password char(20)
                               );""")
        except:
            pass
        # Once all the setup is done and the programming will be executed properl
y
        print("Successfully connected to the database")
        time.sleep(1.5)
        break
    except sql.err.OperationalError:
        print('INCORRECT PASSWORD')
```

```
def hazh(x):  # Hashing function that is used to store passowrds
    sume = sumo = suma = pr4 = pr3 = 0
    ods = evs = prall = 1
    for i in range(0, len(x), 2):
        sume += ord(x[i])
    for i in range(1, len(x), 2):
        sumo += ord(x[i])
    for i in range(len(x)):
        prall *= ord(x[i])+i
    for i in range(0, len(x), 4):
        pr4 += (ord(x[i]) * ord(x[i-1]) * ord(x[i-2]) * ord(x[i-3]))
    for i in range(0, len(x), 3):
        pr3 += (ord(x[i]) * ord(x[i-1]) * ord(x[i-2]))
    for i in range(1, len(x), 2):
        ods *= ord(x[i])
    for i in range(0, len(x), 2):
        evs *= ord(x[i])
    # for i in range(0,len(x),2):
    prs = hex(sume*sumo)[-1:-3:-1]
    suma = (sume+sumo)*len(x) - sume
    suma = hex(suma)[-1:-3:-1]
    if ods > evs:
        oediff = ods-evs
    else:
        oediff = ods + evs
    while not oediff % 16:
        oediff //= 16
    oediff = hex(oediff)[-1:-3:-1]
    while not prall % 16:
        prall //= 16
    prall = hex(prall)[-1:-5:-1]
    pr43 = hex(pr4 % pr3)[-1:-3:-1]
    result = prall+suma+oediff+prs+pr43

    return result

def clear_shell():  # A function that clears the terminal
    if os.name == 'nt':
        os.system('cls')
    elif os.name == 'posix':
        os.system('clear')
    return None

def retrieve_user_list():  # This is used to retrieve the users every time they a
re updated
```

```
    record_tuples = user_cursor.execute('select * from users;')
    record_tuples = user_cursor.fetchall()
    _user = {}
    for i in record_tuples:
        _user[i[0]] = i[1]
    return _user

def retrieve_group_list():  # This is used to retrieve the groups every time they
 are updated
    record_tuples = user_cursor.execute('select * from messaginggroups;')
    record_tuples = user_cursor.fetchall()
    _group = {}
    for i in record_tuples:
        _group[i[0]] = i[1]
    return _group

def create_account():  # Creating an account ie tables for the inbox and outbox
    global user_list
    username = input("Enter the username: ")
    if username in user_list:
        print('This username already exists')
        return create_account()
    else:
        password = hazh(input("Enter your password: "))
        user_cursor.execute(
            f"insert into users values('{username}','{password}')")
        user_cursor.execute(f"""create table {username}_inbox (
                            sender char(20) not null,
                            message text
                            );""")
        user_cursor.execute(f"""create table {username}_outbox (
                            sender char(20) not null,
                            message text
                            );""")
        user_db.commit()
        user_list = retrieve_user_list()
        print("Account created successfully")
        time.sleep(1.5)
    return username

def login(username):  # Auth function
    password = hazh(input("Enter your password: "))
    if not password == user_list[username]:
        print("Incorrect password...")
        return login(username)
```

```
    else:
        return True

def check_user_in_group(user, group):
    user_cursor.execute(f"select * from {group}")
    records = user_cursor.fetchall()
    for i in records:
        if i[0] == user:
            return True
    return False

def send_message(username):  # Sends a message from {username} to {receiever}
    receiver = input("To(username/groupname): ")
    if receiver not in user_list and receiver not in group_list:
        print("This receiver does not exist")
        return send_message()
    else:
        if receiver in group_list and not check_user_in_group(username, receiver)
:
            print("You are not in this group, so you can't send messages in it.")
            time.sleep(1.5)
            return None

        message = input("Message: ")
        user_cursor.execute(
            f"insert into {receiver}_inbox values('{username}','{message}');")
        user_cursor.execute(
            f"insert into {username}_outbox values('{receiver}','{message}');")
        user_db.commit()
        print("Message sent successfully")
        time.sleep(1.5)
        return True

def check_inbox(username):  # To check the inbox of {username}
    # If it is not an individual user and it is a group
    if not username in user_list and username in group_list:
        # Then ask for a password
        password = hazh(input("Enter the password of the group: "))
        # Return the function if the password is incorrect
        if not password == group_list[username]:
            print("Incorrect password...")
            return check_individual_outbox(username)
    messages = user_cursor.execute(f"select * from {username}_inbox;")
    messages = user_cursor.fetchall()
```

```
    if not len(messages):
        print("NO MESSAGES FOUND")
        return None
    for i in messages:
        print(f"{i[0]}: {i[1]}")
        print("\n")
        print("----------------------")
    return None

def check_individual_outbox(username):  # To check the outbox of {username}
    messages = user_cursor.execute(f"select * from {username}_outbox;")
    messages = user_cursor.fetchall()
    for i in messages:
        print(f"{i[0]}: {i[1]}")
        print("\n")
        print("----------------------")
    return None

def erase_inbox(username):  # Erasing all the messages from the inbox of a user
    user_cursor.execute(f"drop table {username}_inbox;")
    user_cursor.execute(f"""create table {username}_inbox (
                            sender char(20) not null,
                            message text
                            );""")
    user_db.commit()
    return None

def create_group(username):  # Creating a group
    global group_list
    groupname = input("Enter the name of the group: ")
    if groupname in group_list:
        print("This group name already exists")
        return create_group(username)
    else:
        # The variable admin isn't really required but I still made it coz why no
t when a statement makes more sense?
        # That statement is the one where the adming is automatically added to th
e group no matter what
        admin = username
        grouppassword = input("Enter the group password: ")
        user_cursor.execute(
            f"insert into messaginggroups values('{groupname}','{grouppassword}')
")
        # Creating a table for the list of users
        user_cursor.execute(f"""create table {groupname}(
```

```
                                    group_member char(20) primary key
                                    );
                                    """)
        # Creating a table for the messages
        user_cursor.execute(f"""create table {groupname}_inbox(
                                    sender char(20) not null,
                                    message text
                                    );
                                    """)
        # Adding the admin to thr group no matter what
        user_cursor.execute(f"insert into {groupname} values('{admin}');")

        # In case the user doesn't input a number for the no_of_group_members
        while True:
            try:
                no_of_group_members = int(
                    input("How many group_members do you want in the group? "))
                print("If you want to stop entering usernames at a point, type 'e
xit'")
                break
            except:
                print("Enter an integer")

        i = 1
        while i <= no_of_group_members:
            group_member = input(f"Enter usename {i}")
            if group_member == 'exit':
                break
            elif not group_member in user_list:
                print("This user does not exist")
            else:
                try:  # Checking if the user has entered a duplicate username
                    user_cursor.execute(
                        f"insert into {groupname} values('{group_member}');")
                    i += 1
                except:
                    print("This user is already in the group")
                else:  # If the user has been added to the table successfully:
                    user_cursor.execute(
                        f"insert into {group_member}_inbox values('{admin}','You
have been added to a group called {groupname} and the password is {grouppassword}
');")

        user_db.commit()
```

```
        # Retrieving the group_list again because a new group has just been creat
ed
        group_list = retrieve_group_list()
        return True

def delete_account(username):  # Deleting the account of the user from this messa
gin service completely
    global user_list
    print('\n\n')
    print("THIS WILL PERMANENTLY DELETE YOUR ACCOUNT FROM THIS SERVICE")
    print("THIS CHANGE WILL BE IRREVERSIBLE")
    print('\n\n')
    print("Are you sure you want to delete the account ?")
    choice = input(
        f"Confirm by typing in '{username}:{user_list[username]}' or type anythin
g else to cancel: ")

    # If the user chooses to delete the account
    if choice == f"{username}:{user_list[username]}":
        # Deleting the record from the users table
        user_cursor.execute(
            f"delete from users where username = '{username}';")
        user_cursor.execute(f"drop table {username}_inbox;")
        user_cursor.execute(f"drop table {username}_outbox;")
        for i in group_list:
            # Deletes the record of the user if they are in the group
            try:
                user_cursor.execute(
                    f"delete from {i} where group_member = '{username}';")
            except:
                pass
        user_list = retrieve_user_list()
        user_db.commit()
        return True

    # If the user chooses not to delete the account
    else:
        return False

def view_all_users(username):  # Viewing a list of all users except the current u
ser themself
    print("\n\n")
    a = 1
    for i in user_list:
        if not i == username:
```

```
            print(f"{a}. {i}")
            a+=1
    return None

def login_menu(username):  # Menu of all tasks that a user can perform
    clear_shell()
    print("These are all the tasks you can perform")
    print("1.Send a message")
    print("2.Check your individual inbox")
    print("3.Log out")
    print("4.Erase all messages in your inbox")
    print("5.Check your outbox")
    print("6.Create a group")
    print("7.Check your group inbox")
    print("8.View a list of all users")
    print("9.Delete account")
    choice = input("Enter your choice: ")
    if choice == '1':
        send_message(username)
    elif choice == '2':
        check_inbox(username)
    elif choice == '3':
        return None
    elif choice == '4':
        erase_inbox(username)
    elif choice == '5':
        check_individual_outbox(username)
    elif choice == '6':
        create_group(username)
    elif choice == '7':
        groupname = input("Enter the name of the group:")
        check_inbox(groupname)
    elif choice == '8':
        view_all_users(username)
    elif choice == '9':
        if delete_account(username):
            print("Account successfully deleted...")
            time.sleep(1.5)
            return None
        else:
            print("Deletion aborted")
    else:
        print("Enter a valid option")
    input("Press enter to continue...")
    return login_menu(username)
```

```
def Menu():  # Main/Initial Menu
    clear_shell()
    print("Choose what you want to do: ")
    print("1.Create an account")
    print("2.Login")
    print("3.Exit")
    response = input("Enter your response: ")
    if response == "1":
        login_menu(create_account())
    elif response == "2":
        username = input("Enter your username: ")
        if username not in user_list:
            print("This username doesn't exist...")
            print("Try creating an account: ")
        else:
            if (login(username)):
                login_menu(username)
    elif response == "3":
        user_cursor.close()
        user_db.close()
        exit()
    else:
        print("Invalid response")
    input("Press Enter to continue...\n\n")
    Menu()

if __name__ == "__main__":
    user_list = retrieve_user_list()
    group_list = retrieve_group_list()
    Menu()
```

OUTPUT:

Choose what you want to do:

1.Create an account

2.Login

3.Exit

Enter your response: 2

Enter your username: SPARTACUS

Enter your password: Nothing

These are all the tasks you can perform

1.Send a message

2.Check your individual inbox

3.Log out

4.Erase all messages in your inbox

5.Check your outbox

6.Create a group

7.Check your group inbox

8.View a list of all users

9.Delete account

Enter your choice: 1

To(username/groupname): lingling

Message: Hello how are you?

Message sent successfully

Press enter to continue...

These are all the tasks you can perform

1.Send a message

2.Check your individual inbox

3.Log out

4.Erase all messages in your inbox

5.Check your outbox

6.Create a group

7.Check your group inbox

8.View a list of all users

9.Delete account

Enter your choice: 3

Press Enter to continue...

Choose what you want to do:

1.Create an account

2.Login

3.Exit

Enter your response: 2

Enter your username: lingling

Enter your password: 40hrs

These are all the tasks you can perform

1.Send a message

2.Check your individual inbox

3.Log out

4.Erase all messages in your inbox

5.Check your outbox

6.Create a group

7.Check your group inbox

8.View a list of all users

9.Delete account

Enter your choice: 2

SPARTACUS: Hello how are you?

Press enter to continue...

These are all the tasks you can perform

1.Send a message

2.Check your individual inbox

3.Log out

4.Erase all messages in your inbox

5.Check your outbox

6.Create a group

7.Check your group inbox

8.View a list of all users

9.Delete account

Enter your choice: 6

Enter the name of the group: Group1

Enter the group password: Group123

How many group_members do you want in the group? 2

If you want to stop entering usernames at a point, type 'exit'

Enter usename 1SPARTACUS

Enter usename 2Davie

This user does not exist

Enter usename 2UGN

This user does not exist

Enter usename 2exit

Press enter to continue...

Choose what you want to do:

1.Create an account

2.Login

3.Exit

Enter your response: 2

Enter your username: SPARTACUS

Enter your password: Nothing

These are all the tasks you can perform

1.Send a message

2.Check your individual inbox

3.Log out

4.Erase all messages in your inbox

5.Check your outbox

6.Create a group

7.Check your group inbox

8.View a list of all users

9.Delete account

Enter your choice: 2

lingling: You have been added to a group called UGN2 and the password is Nothing

---

lingling: You have been added to a group called Group1 and the password is Group123

Press enter to continue...

These are all the tasks you can perform

1.Send a message

2.Check your individual inbox

3.Log out

4.Erase all messages in your inbox

5.Check your outbox

6.Create a group

7.Check your group inbox

8.View a list of all users

9.Delete account

Enter your choice: 1

To(username/groupname): Group1

Message: Hello everyone!

Message sent successfully

Press enter to continue...

Choose what you want to do:

1.Create an account

2.Login

3.Exit

Enter your response: 2

Enter your username: lingling

Enter your password: 40hrs

These are all the tasks you can perform

1.Send a message

2.Check your individual inbox

3.Log out

4.Erase all messages in your inbox

5.Check your outbox

6.Create a group

7.Check your group inbox

8.View a list of all users

9.Delete account

Enter your choice: 7

Enter the name of the group:Group1

Enter the password of the group: Group123

SPARTACUS: Hello everyone!

Press enter to continue...

These are all the tasks you can perform

1.Send a message

2.Check your individual inbox

3.Log out

4.Erase all messages in your inbox

5.Check your outbox

6.Create a group

7.Check your group inbox

8.View a list of all users

9.Delete account

Enter your choice: 8

1. SPARTACUS

Press enter to continue...

These are all the tasks you can perform

1.Send a message

2.Check your individual inbox

3.Log out

4.Erase all messages in your inbox

5.Check your outbox

6.Create a group

7.Check your group inbox

8.View a list of all users

9.Delete account

Enter your choice: 9

THIS WILL PERMANENTLY DELETE YOUR ACCOUNT FROM THIS SERVICE

THIS CHANGE WILL BE IRREVERSIBLE

Are you sure you want to delete the account ?

Confirm by typing in 'lingling:810b66c2e782' or type anything else to cancel: lingling:810b66c2e782

Account successfully deleted...

Press Enter to continue...

Choose what you want to do:

1.Create an account

2.Login

3.Exit

Enter your response: 3