Satish Chandra Memorial School
Affiliated to CBSE - DELHI(10+2),AFFILIATION NO.2430102
(Sr. Secondary - Science, Arts and Commerce)
(ISO 9001:2008 Certified Organisation)

The Project Report On

-- CHARITY MANAGEMENT --

For AISSCE 2023-24 Examination

Submitted by:

NAME → Bedanta Dey

CLASS → XII - C

CBSE Roll No. →

Under the guidance of

Sourish Sen, HOD PGT Computer Science

CERTIFICATE

This is to certify that the project/dissertation entitled “Charity Management” is a bonafide work done by Bedanta Dey of class XII-C, Session 2023 – 24 in partial fulfilment of CBSE’s AISSCE examination, and has been carried out under my direct supervision and guidance. This report or a similar report on the topic has not been submitted for any other examination and does not form a part of in any other course undergone by the candidate.

…… ……..

Signature of Student

Bedanta Dey

Signature of Teacher

Sourish Sen

PGT Comp. Sc.

Place: Chakdaha

Date: ……

Signature of the External Examiner

ACKNOWLEDGEMENT

I would like to acknowledge all those without whom this project would not have been successful. Firstly, I would wish to thank our Computer Science teacher Mr. Sourish Sen Sir, who guided me throughout the project and gave his immense support. He made us understand how to successfully complete this project and without him, the project would not have been complete.

This project has been a source to learn and bring our theoretical knowledge to the real-life world. So, I would really acknowledge his help and guidance for this project.

I would also like to thank my parents who have always been there whenever needed.

Once again, thanks to everyone for making this project successful.

Bedanta Dey

Class – XII-C

CONTENTS

INTRODUCTION

Charity Donation Management System (CDMS) is designed to streamline the management of charitable donations and donors. In today's fast-paced world, effective management of charitable resources is crucial. This software offers an intuitive and efficient solution for organizations and individuals involved in philanthropic activities.

This software is packed with features like User Authentication, Donor Management, Donation Tracking, Data Integrity with MySQL Backend and is very simple and usable.

Charity Donation Management System empowers organizations to efficiently manage donors and donations, fostering transparency, accountability, and ultimately, making a positive impact on the lives of those in need. Whether you are a charitable organization, fundraiser, or donor, our system is designed to simplify the process of managing and tracking charitable contributions, enabling you to focus on what truly matters – making a difference in the world.

OBJECTIVE & SCOPE OF THE PROJECT

The objective of this project is to let the students apply the programming knowledge into a real- world situation/problem and exposed the students how programming skills helps in developing a good software.

1. Write programs utilizing modern software tools.

2. Apply object-oriented programming principles effectively when developing small to medium sized projects.

3. Write effective procedural code to solve small to medium sized problems.

4. Students will demonstrate a breadth of knowledge in computer science, as exemplified in the areas of systems, theory and software development.

5. Students will demonstrate the ability to conduct research or applied Computer Science projects, requiring writing and presentation skills that exemplify a scholarly style in computer science.

PROBLEM DEFINITION & ANALYSIS

During this project, I have faced a few issues; like learning Python-MySQL connectivity properly, and a few other problems.

One of the main problems was to arrange the UI sequence. For this problem, I drew a flowchart that showed through what steps a user can/will go to use the program, starting with login and ending with getting an output.

I was also confused about how to unify the donors and donations, for which I used a common scope, the "`DonorID`". This value can be used to identify both donors and donations, and can be used across tables.

I also found it kind of "unsatisfactory" to clump both functions and the main program in one file. For this, I made separate files for the program and the functions. I named the function-only file as "`sqlfunctions.py`" and it was used by importing it using "`import`".

I also learned about placeholder values like "%s", which helps to take values, just like arguments.

These were some of the problems I faced whilst making this project and my ways of tackling those issues. Overall, I learnt a lot of new things through this project.

SYSTEM IMPLEMENTATION

4.1 THE HARDWARE USED:

While the system was being developed, the hardware was:

Laptop computer with an Intel Celeron N4000 Processor and Mesa Intel UHD 600 (Integrated) Graphics with 4GB of RAM and 1TB of HDD Storage.

4.2 THE SOFTWARE USED:

While the system was being developed, the software consisted of:

- Microsoft Windows 11 and Ubuntu as operating systems
- Python 3.12 as front-end development environment
- Visual Studio Code as IDE
- MySQL as back-end development environment
- Microsoft Office Word 2021 for creating this documentation

SYSTEM DESIGN & DEVELOPMENT

5.1 DATABASE DESIGN:

Database Name: `charity`

Tables under database: `donations` | `users` | `donors`

`donations` Table:

Field	Type	Null	Key	Default	Extra
donor_id	int	YES		NULL	
amount	int	YES		NULL	
donation_date	date	YES		NULL	

`users` Table:

Field	Type	Null	Key	Default	Extra
username	varchar(255)	YES		NULL	
password	varchar(255)	YES		NULL	

`donors` Table:

Field	Type	Null	Key	Default	Extra
name	varchar(255)	YES		NULL	
email	varchar(255)	YES		NULL	
donor_id	int	YES		NULL	

5.2 MENU DESIGN:

Start Screen:

```
Welcome to CDMS!
Developed by Bedanta Dey | V.0.0.1
Please proceed with appropriate inputs to continue.

Enter an input:
1. Login
2. Create an Account
3. Exit
Enter your choice:
```

CDMS Menu:

```
CDMS Menu:
1. Add Donor
2. Make Donation
3. View Donors
4. View Donations
5. Delete Donor
6. Logout
Enter your choice:
```

5.3 SOURCE CODING:

... PTO

main.py →

```
import mysql.connector as sqltor
import sqlfunctions as sqlf

mycon = sqltor.connect(host="localhost", user="root",
password="123456", database="charity")
cursor = mycon.cursor()

# Welcome Screen
print("Welcome to CDMS!")
print("Developed by Bedanta Dey | V.0.0.1")
print("Please proceed with appropriate inputs to continue.")
print("")

while True:
    print("Enter an input:")
    print("1. Login")
    print("2. Create an Account")
    print("3. Exit")
    choice = input("Enter your choice: ")

    if choice == "1":
        # User Login
        username = input("Enter your username: ")
        password = input("Enter your password: ")
        user = sqlf.check_credentials(username, password)
        if user:
            print("Login successful!")
            print("")
```

```
            print("")
            while True:
                print("\nCDMS Menu:")
                print("1. Add Donor")
                print("2. Make Donation")
                print("3. View Donors")
                print("4. View Donations")
                print("5. Delete Donor")
                print("6. Logout")
                option = input("Enter your choice: ")

                if option == "1":
                    name = input("Enter donor's name: ")
                    email = input("Enter donor's email: ")
                    donor_id = input("Set donor ID: ")
                    sqlf.add_donor(name, email, donor_id)
                elif option == "2":
                    donor_id = int(input("Enter donor ID: "))
                    amount = float(input("Enter donation amount: "))
                    donation_date = input("Enter donation date
(YYYY-MM-DD): ")
                    sqlf.make_donation(donor_id, amount,
donation_date)
                elif option == "3":
                    sqlf.view_donors()
                elif option == "4":
                    sqlf.view_donations()
                elif option == "5":
                    donor_id = input("Enter donor's ID: ")
                    sqlf.delete_donor(int(donor_id))
                elif option == "6":
```

```
                    print("Logged out. Goodbye!")
                    print("")
                    print("")
                    break
                else:
                    print("Invalid choice. Please choose a valid
option.")
                    print("")
                    print("")
        else:
            print("Invalid username or password. Please try again.")
            print("")
            print("")
    elif choice == "2":
        # User Registration
        username = input("Enter a username: ")
        password = input("Enter a password: ")
        query = "SELECT * FROM users WHERE username = %s"
        cursor.execute(query, (username,))
        existing_user = cursor.fetchone()

        if existing_user:
            print("Username already taken. Please choose another
username.")
            print("")
            print("")
        else:
            sqlf.create_user(username, password)
            print("Account created successfully! You can now log
in.")
            print("")
```

```
            print("")
    elif choice == "3":
        print("Exiting. Goodbye!")
        break
    else:
        print("Invalid choice. Please choose a valid option.")
        print("")
        print("")

mycon.close()
```

sqlfunctions.py →

```
"""All functions for a smooth sailing charity donation experience"""
"""This is NOT an external library. This was made by me."""
import mysql.connector as sqltor

mycon = sqltor.connect(host="localhost", user="root",
password="123456", database="charity")
cursor = mycon.cursor()

def create_user(username, password):
    '''Creates a USER for the CDMS'''
    query = "INSERT INTO users (username, password) VALUES (%s, %s)"
    values = (username, password)
    cursor.execute(query, values)
    mycon.commit()

def check_credentials(username, password):
```

```
    '''Validates username and password against the database'''
    query = "SELECT * FROM users WHERE username = %s AND password =
%s"
    values = (username, password)
    cursor.execute(query, values)
    user = cursor.fetchone()
    return user


def add_donor(name, email, donor_id):
    '''Adds a donor'''
    query = "INSERT INTO donors (name, email, donor_id) VALUES (%s,
%s, %s)"
    values = (name, email,donor_id)
    cursor.execute(query, values)
    mycon.commit()
    print("Donor added successfully!")
    print("")
    print("")


def make_donation(donor_id, amount, donation_date):
    '''Makes a donation'''
    query = "INSERT INTO donations (donor_id, amount, donation_date)
VALUES (%s, %s, %s)"
    values = (donor_id, amount, donation_date)
    cursor.execute(query, values)
    mycon.commit()
    print("Donation recorded successfully!")
    print("")
    print("")


def delete_donor(donor_id):
```

```
    '''Deletes a donor'''
    delete_donations_query = "DELETE FROM donations WHERE donor_id =
%s"
    cursor.execute(delete_donations_query, (donor_id,))
    mycon.commit()

    delete_donor_query = "DELETE FROM donors WHERE donor_id = %s"
    cursor.execute(delete_donor_query, (donor_id,))
    mycon.commit()
    print(f"User '{donor_id}' deleted successfully.")

# DISPLAY FUNCtions
def view_donors():
    '''Views donors'''
    query = "SELECT * FROM donors"
    cursor.execute(query)
    donors = cursor.fetchall()
    print("")
    print("")
    for x in donors:
        for y in x:
            print(y, end="   ")
        print()
    print("")

def view_donations():
    '''Views donations'''
    query = "SELECT * FROM donations"
    cursor.execute(query)
    donations = cursor.fetchall()
```

```
    print("")
    print("")
    for x in donations:
        for y in x:
            print(y, end="   ")
        print()
    print("")
```

5.3 I/O SCREEN:

User opens CDMS, creates an account and logs in using said account

```
Welcome to CDMS!
Developed by Bedanta Dey | V.0.0.1
Please proceed with appropriate inputs to continue.

Enter an input:
1. Login
2. Create an Account
3. Exit
Enter your choice: 2
Enter a username: demoaccount1
Enter a password: demopassword
Account created successfully! You can now log in.


Enter an input:
1. Login
2. Create an Account
3. Exit
Enter your choice: 1
Enter your username: demoaccount1
Enter your password: demopassword
Login successful!
```

User adds a donor named “Generous Person” [ID 7]

```
CDMS Menu:
1. Add Donor
2. Make Donation
3. View Donors
4. View Donations
5. Delete Donor
6. Logout
Enter your choice: 1
Enter donor's name: Generous Person
Enter donor's email: geneper@test.mail
Set donor ID: 7
Donor added successfully!
```

User sets donation amount for ID 7

```
CDMS Menu:
1. Add Donor
2. Make Donation
3. View Donors
4. View Donations
5. Delete Donor
6. Logout
Enter your choice: 2
Enter donor ID: 7
Enter donation amount: 50000
Enter donation date (YYYY-MM-DD): 2023-11-11
Donation recorded successfully!
```

User views all donors (Format: Name, Email, ID)

```
CDMS Menu:
1. Add Donor
2. Make Donation
3. View Donors
4. View Donations
5. Delete Donor
6. Logout
Enter your choice: 3


Lorem Ipsum Guy   loremipsum@dolor.sit   1
Generous Person   geneper@test.mail   7
```

User views all donations (Format: ID, Amount, Date)

```
CDMS Menu:
1. Add Donor
2. Make Donation
3. View Donors
4. View Donations
5. Delete Donor
6. Logout
Enter your choice: 4


1   54000   2023-11-11
7   50000   2023-11-11
```

User deletes details of ID 1

```
CDMS Menu:
1. Add Donor
2. Make Donation
3. View Donors
4. View Donations
5. Delete Donor
6. Logout
Enter your choice: 5
Enter donor's ID: 1
User '1' deleted successfully.
```

None of ID 1's details can be found after deletion

```
CDMS Menu:
1. Add Donor
2. Make Donation
3. View Donors
4. View Donations
5. Delete Donor
6. Logout
Enter your choice: 3


Generous Person   geneper@test.mail   7
```

```
CDMS Menu:
1. Add Donor
2. Make Donation
3. View Donors
4. View Donations
5. Delete Donor
6. Logout
Enter your choice: 4


7   50000   2023-11-11
```

USER MANUAL

6.1 How to install the software

Create a separate folder and copy the contents (`main.py` and `sqlfunctions.py`) to that folder. Then run main.py using a Python interpreter. In Windows, one may double-click to execute the program. In Linux, one may open the folder in terminal and type in "`python3 ./main.py`" (without quotes) to run the program. **MAKE SURE BOTH `main.py` AND `sqlfunctions.py` FILES ARE IN THE SAME DIRECTORY.**

6.2 Working with software

To work in optimum condition with this software, one must keep in mind these following requirements:

Recommended hardware requirements:

- 2GB RAM or more
- 1GB of available storage or more
- Working keyboard

Recommended software requirements:

- Windows 10/11 or any Linux-based distributions or Mac OS 10.14+
- Python 3.8+ and required libraries like `mysql-connector`
- MySQL 8.0

One must also create the required database "charity" and the tables "donors", "donations" and "users" beforehand to avoid errors.

BIBLIOGRAPHY

1. Arora, Sumita. Computer Science with Python Class XII. 2023.

2. MySQL Tutorial. W3Schools. https://www.w3schools.com/MySQL/default.asp