

Table of Contents

Sr. No.	Heading	Page No.
1	Abstract	3
2	Introduction of the Topic	4
3	Functions to be Developed	4
4	Literature Study of the Topic	5-9
5	Data Flow Diagrams	10-11
6	Software Flowchart	12
7	Source Code	13-26
8	Database	27-33
9	Output: Implementation Results	34-38
10	Advantages & Disadvantages	39-40
11	Applications	41
12	Conclusion	42
13	References	43

Numismatic Gallery Management System

Abstract

In this project, we will try to develop an interface to facilitate the posting and fetching of information about currency which is part of our collection. The idea is to develop a program that will allow viewers of the collection to look for the currency they are interested in by entering the country it belongs to. In case the collector does not have any currency of that country, it will allow the user to add information about it from his side by filling details in fields like year, description, country, denomination and composition. In case the currency the user is looking for is present in the collection, the user will be presented with a choice whether he is looking for a coin or a banknote. Based on the user input, the interface will display all the relevant currency and the user will be able to search for a particular currency by entering its denomination. The user will be able to view the information of the currency and whether it is available for trading or not.

In this project, we will also allow the users to create customer accounts, where they can see the cart subtotal and the items, if they wish to buy currency which is available for sale.

The essence of this project is to create a fully functional interactive collection display gallery as seen on the instagram page owned by us [@numismatic_gallery](#).

Introduction of the Numismatic Gallery Management System

We will make this project by using Python and mySQL interface. It will be a menu-driven program, in which we will use different user-defined functions for different actions that the user wants to perform. Some of the actions that the user will be able to perform are:

- | | |
|------------------|------------------|
| ★ View details | ★ Add details |
| ★ Delete details | ★ Admin mode |
| ★ User login | ★ Update details |
| ★ View Cart | ★ Exit |

Some of the functions that we will develop for the same purpose are as follows:

Functions developed:

country(): This function will look at whether the currency of the entered country exists in the collection or not.

add_coin(): This function will help the user to enter information about the currency coin that is not yet part of the collection.

add_note(): This function will help the user to enter information about the currency note that is not yet part of the collection.

delete_coin(): This function will help the user to delete information about the currency coin which may be incorrect.

delete_note(): This function will help the user to delete information about the currency coin which may be incorrect.

accounts(): This function will help in managing the customer accounts.

*Note: The final code may vary from the system proposed in synopsis, depending on the issues that will be faced in future.

Literature Study of Numismatic Gallery Management System

1. Introduction

1.1 Purpose: The purpose of this document is to build an online system to manage the collection of coins and banknotes and to ease the collection management. The users/collectors must be able to post and fetch the information about an item. Also, users can create their accounts and login to them to see the contents of their shopping cart in case they want to buy from Numismatic Gallery.

1.2 Intended Audience: This project is a prototype for the collection management system and will be available worldwide. The targeted audience are the collectors of numismatic items, who can fetch information such as year of mint, composition material, availability, design, estimated value, etc. of the collectible they possess or buy the item they wish for.

1.3 Project scope: The purpose of this project is to create a convenient and easy-to-use application for collectors, searching for information or trying to buy items. Above all, we hope to provide a comfortable user experience along with the best pricing available. The Numismatic Gallery Management System will allow collectors to:

- Store and manage detailed information about their coin collection.
- Securely browse their collection through a login system.
- Add desired coins to a shopping cart for potential purchase.
- Complete transactions through the integrated shop feature.

1.4 References: The reference for user interface for this software may be taken from the websites: <https://multiverseweb.github.io/coinshop/> and <https://multiverseweb.github.io/login/>.

1.5 Definitions, Acronyms, and Abbreviations:

- Python: The programming language used for software development.
- MySQL: The relational database management system used for data storage.
- SRS: Software Requirement Specification.

2. Overall Description

2.1 Product Perspective: The Coin Collector Database Management System is a standalone web-based application. It interfaces with a MySQL database to store and retrieve coin collection and shopping cart data. The project database system stores the following information:

- Item Details: It includes the information of collectibles that are already available for sale and users can fetch the information about these items or can buy them too.

- Customer Details: It includes the customer account details such as customer name, password and items in cart.

2.2 Product Features:

2.2.1 User Authentication:

- Users can register using their name and password.
- Secure login mechanism for accessing the system.

2.2.2 Coin Collection Management:

- Users can add, edit, and delete coin entries.
- Each coin entry includes details such as coin name, year, country, condition, and description.
- Organise collections based on attributes (e.g., year, country, condition).

2.2.3 Shopping Cart Management:

- Users can add and manage items in their shopping cart.
- Cart contents can be reviewed and modified.
- Users can complete purchases through the shopping cart.

2.2.4 Shop:

- Displays available coins for sale.
- Listings include coin name, year, country of origin, price, and availability.

2.3 User Classes and Characteristics:

- Collectors: Registered users who can manage their coin collections and make purchases.
- Admin: System administrators responsible for managing user accounts and shop listings.

2.3 Operating Environment: The operating environment for the system is as follows:

- Distributed Database
- Client/Server system
- Operating System: Windows
- Database: MySQL Database
- Backend: Python

3. Functional Requirements

Detailed functional requirements are specified in Section 2.2.

4. Non-Functional Requirements

4.1 Performance Requirements:

- The system shall provide a responsive user experience.
- Database operations should be optimised for minimal latency.

4.2 Security Requirements

- User data and transactions shall be securely encrypted.
- Authentication shall prevent unauthorised access.
- Data backup and recovery mechanisms shall be implemented.

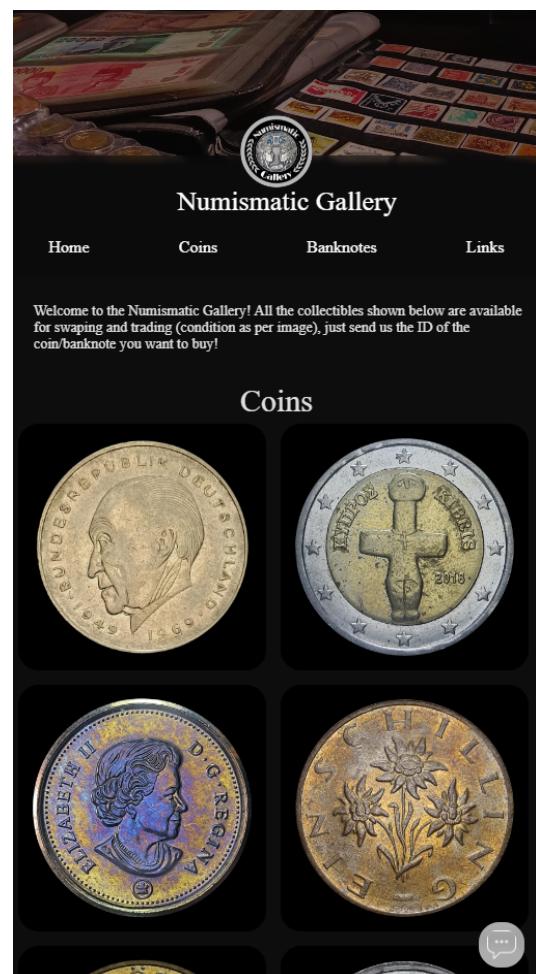
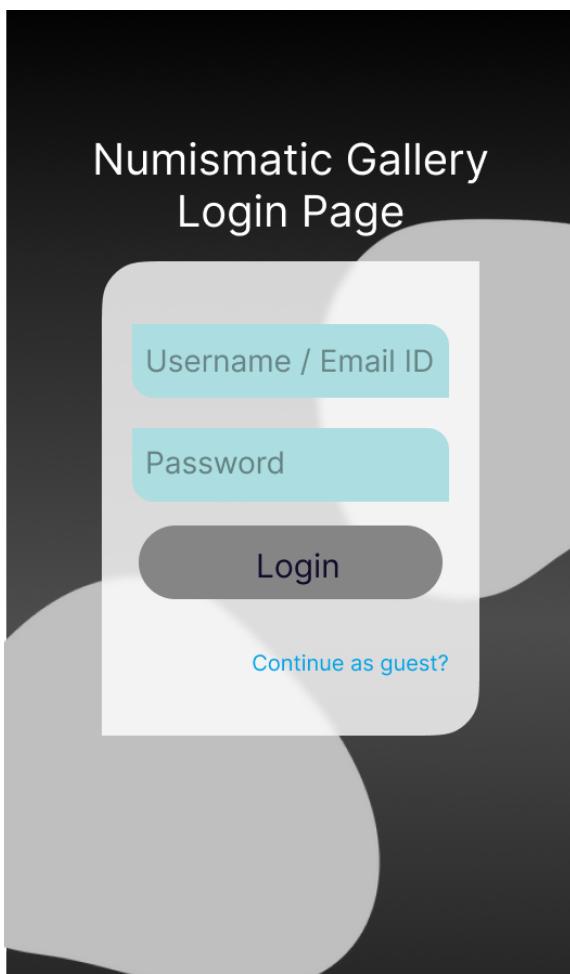
4.3 Usability Requirements

- The user interface shall be intuitive and user-friendly.

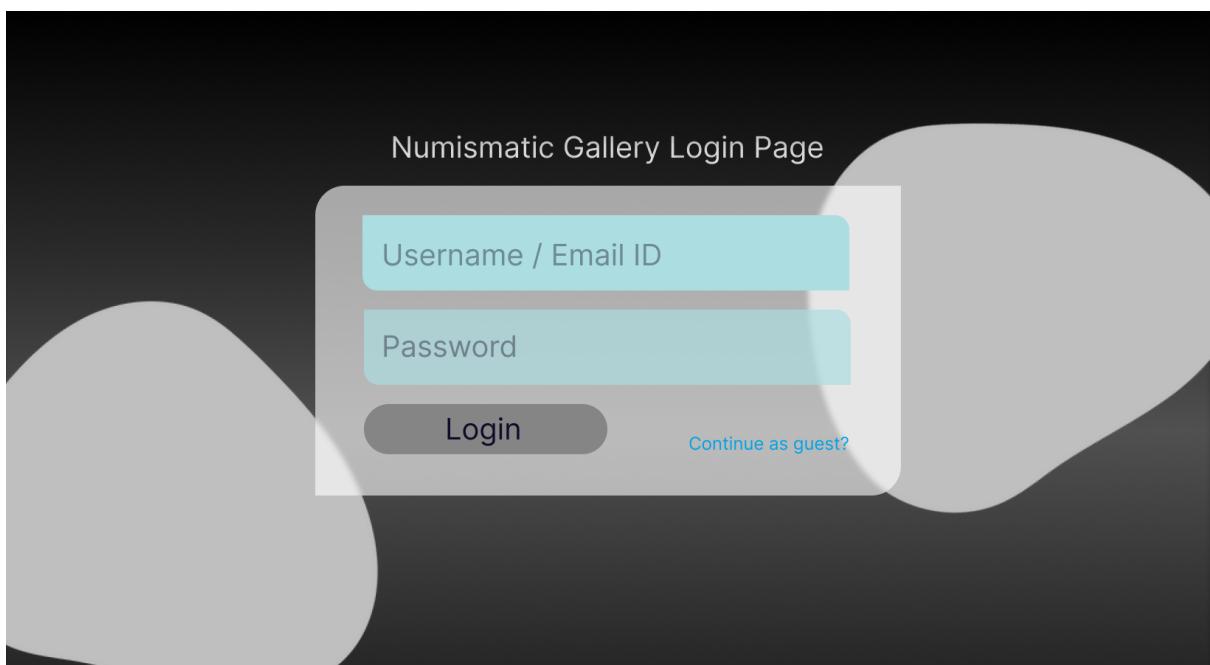
5. Methodology Used

5.1 User Interface

Mobile View



Desktop View





5.2 Hardware Interface:

- A browser that supports HTML, CSS and JavaScript.

5.3 Performance:

- The application shall load quickly and respond to user interactions in a timely manner.
- The map rendering shall be smooth, even on devices with lower processing power.

5.4 Open Source:

- This should be an open source web software in which users should be able to copy the information they want.

5.5 Software Interfaces:

Software Used	Description
Operating System	We have chosen the Windows operating system for its supporting environment and its user-friendliness.
Database	To save records, we have chosen a MySQL database.
Backend Language	We have chosen Python for its more interactive support.

5.6 Communication Interface: The project supports all types of web browsers and devices.

6. Constraints

6.1 Data Storage:

- Dependent on scalable databases for user accounts and interactions, adhering to privacy regulations.

6.2 Internet Connectivity:

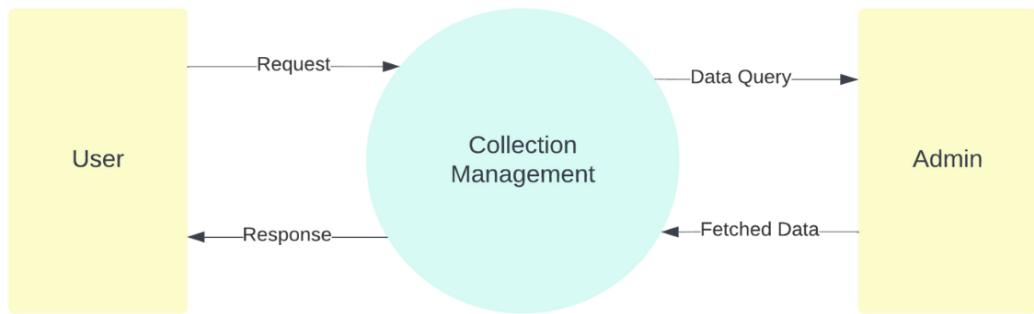
- Platform requires an active internet connection for access.

7. Conclusion

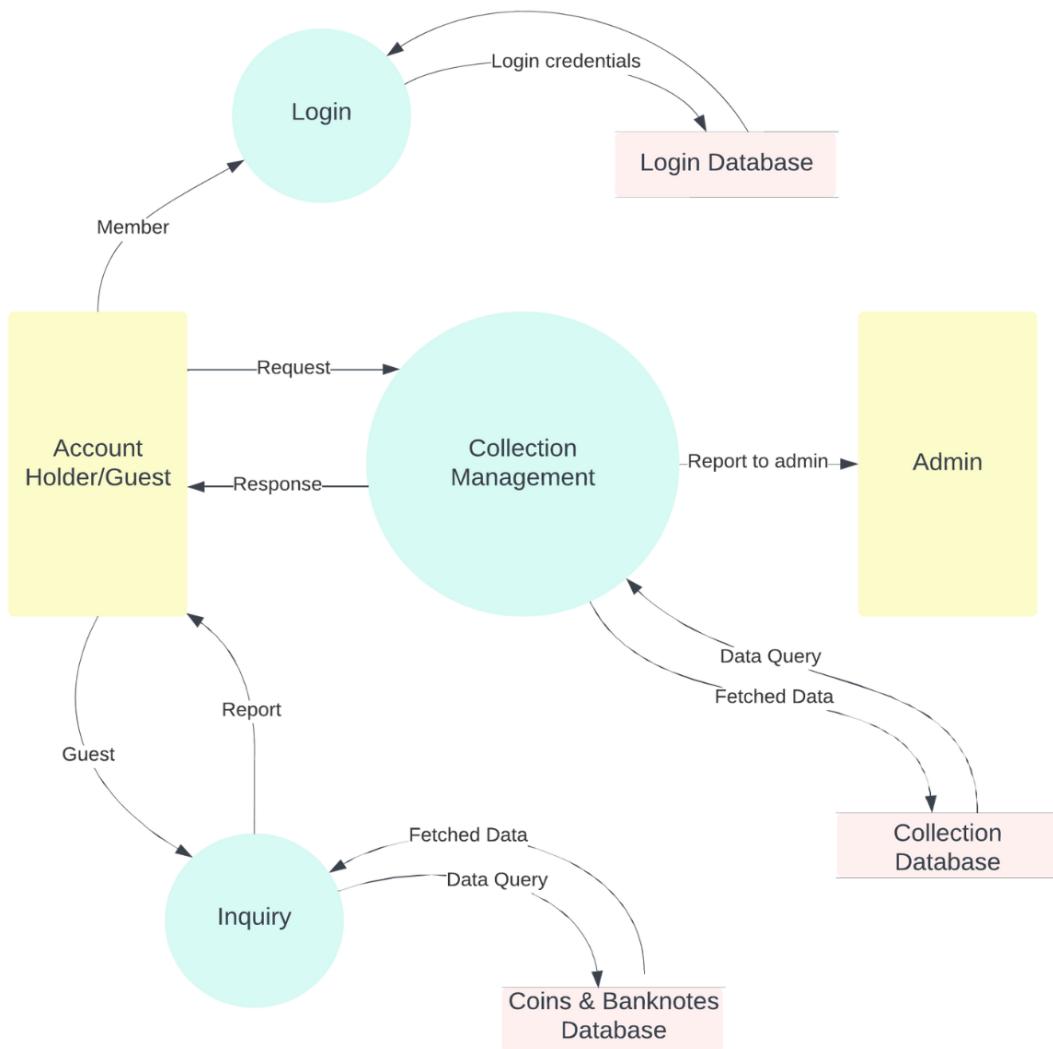
This Software Requirement Specification (SRS) document serves as a comprehensive guide for the development of the Numismatic Gallery Management System, ensuring that all specified requirements are addressed. Detailed requirements ensure a secure, accessible, and user-friendly environment, supported by ongoing maintenance and future enhancements.

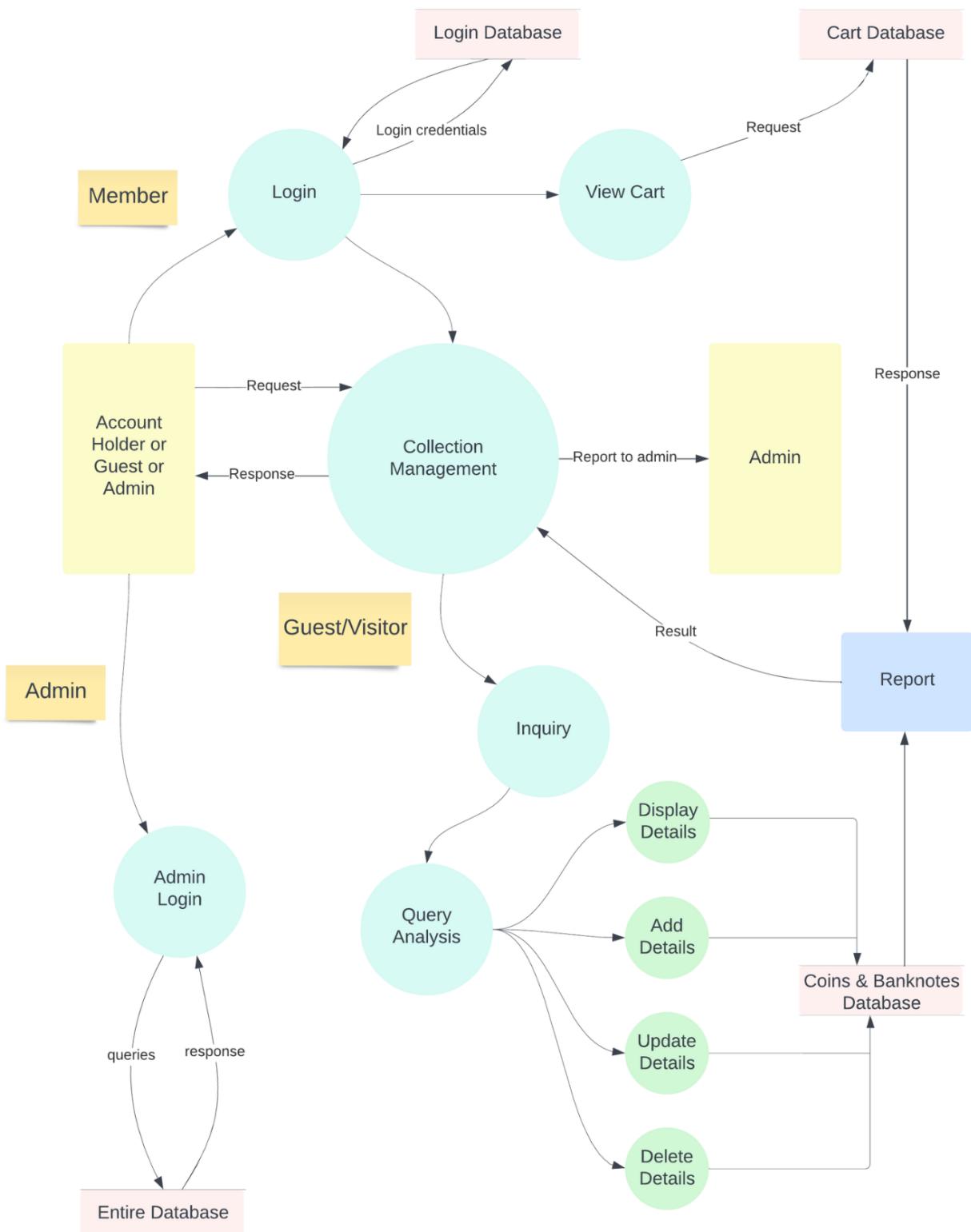
Data Flow Diagrams

Numismatic Gallery Management System- DFD Level 0
20-09-2023

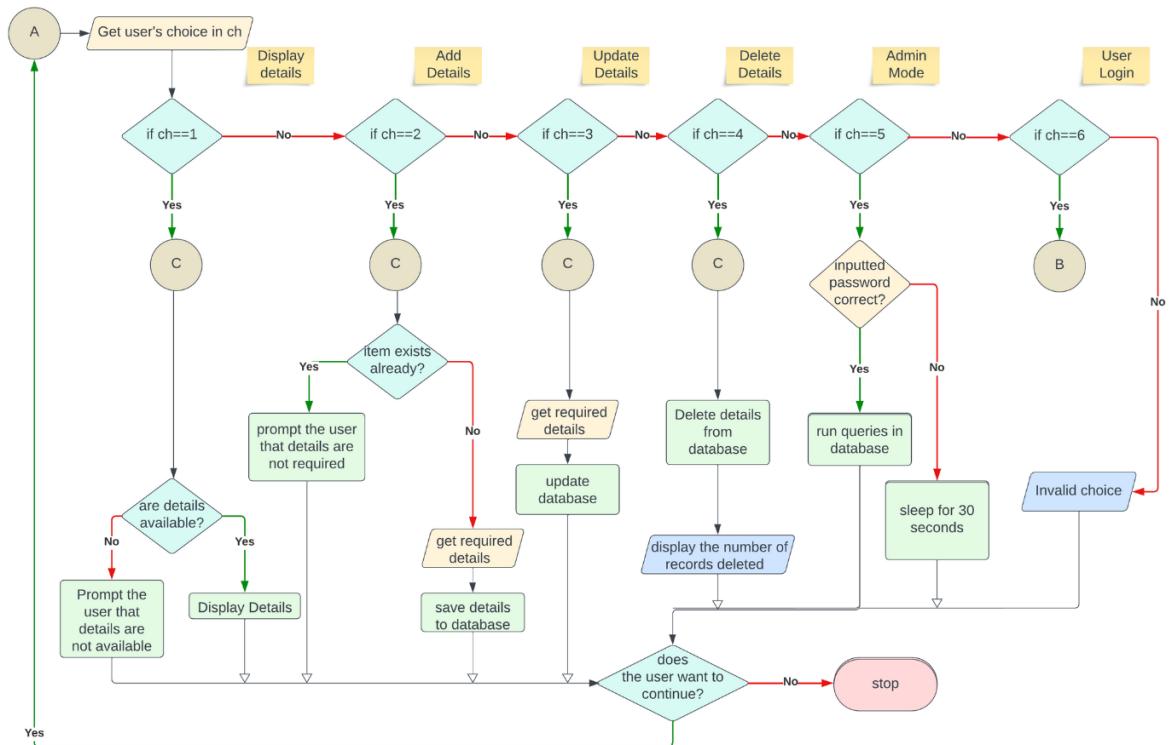
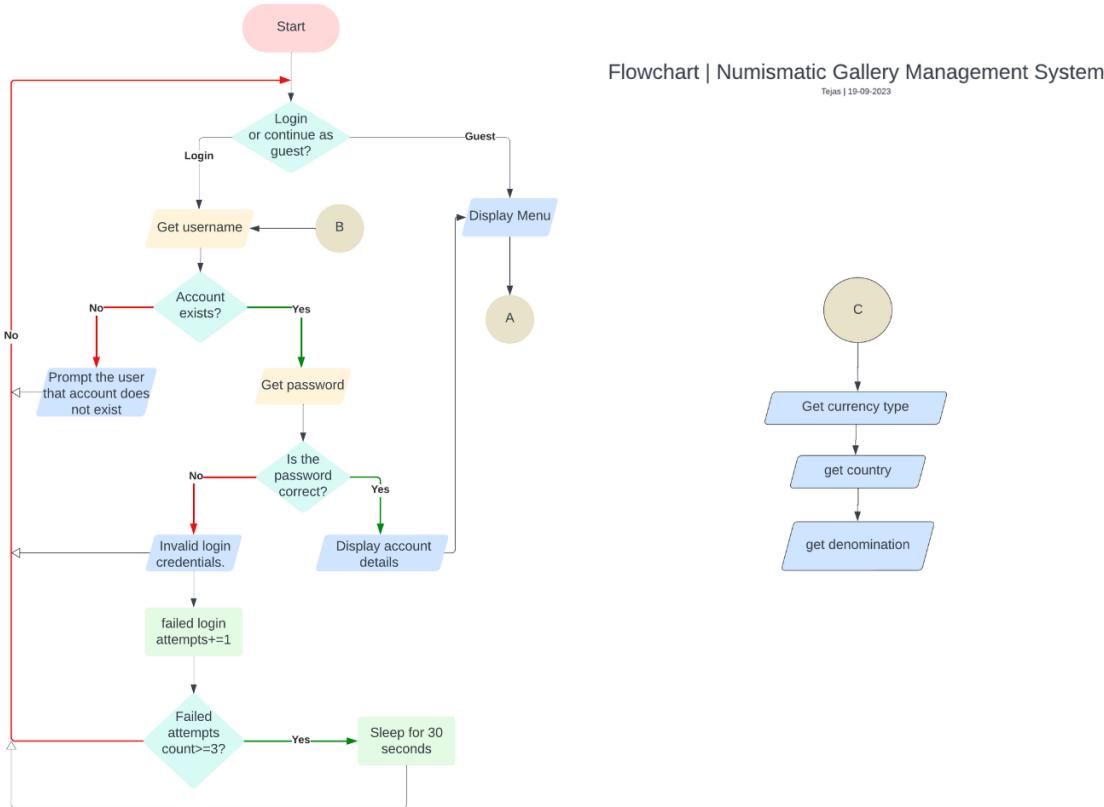


Numismatic Gallery Management System- DFD Level 1
20-09-2023





Flowchart



Source Code

```

    elif col.lower()=='material':
        new=input("Enter new material:")
        cursor2.execute("update coin set material='{}' where
country='{}' and denomination={}".format(new,c.lower(),v))
    elif col.lower()=='diameter':
        new=input("Enter new diameter: ")
        cursor2.execute("update coin set diameter='{}' where
country='{}' and denomination={}".format(new,c.lower(),v))
    elif col.lower()=='thickness':
        new=input("Enter new thickness:")
        cursor2.execute("update coin set thickness='{}' where
country='{}' and denomination={}".format(new,c.lower(),v))
    else:
        print("No such field found.")
    global A
    A=False
    mycon2.commit()
#####
#####                                     #update note details
def update_n(col,c,v):
    if col.lower()=='country':
        new=input("Enter new country name:")
        cursor2.execute("update note set country='{}' where
country='{}' and denomination={}".format(new,c.lower(),v))
    elif col.lower()=='name':
        new=input("Enter new name:")
        cursor2.execute("update note set name='{}' where country='{}'
and denomination={}".format(new,c.lower(),v))

    elif col.lower()=='denomination':
        new=int(input("Enter new denomination:"))
        cursor2.execute("update note set denomination={} where
country='{}' and denomination={}".format(new,c.lower(),v))
    elif col.lower()=='year':
        new=input("Enter new year:")
        cursor2.execute("update note set year='{}' where country='{}'
and denomination={}".format(new,c.lower(),v))
    elif col.lower()=='front side':
        new=input("Enter new details of front side:")
        cursor2.execute("update note set front='{}' where country='{}'
and denomination={}".format(new,c.lower(),v))
    elif col.lower()=='back side':
        new=input("Enter new details of back side:")

```

```

        cursor2.execute("update note set back='{}' where country='{}'
and denomination={}".format(new,c.lower(),v))
    elif col.lower()=='material':
        new=input("Enter new material:")
        cursor2.execute("update note set material='{}' where
country='{}' and denomination={}".format(new,c.lower(),v))
    elif col.lower()=='length':
        new=input("Enter new length: ")
        cursor2.execute("update note set length='{}' where country='{}'
and denomination={}".format(new,c.lower(),v))
    elif col.lower()=='breadth':
        new=input("Enter new breadth:")
        cursor2.execute("update note set breadth='{}' where
country='{}' and denomination={}".format(new,c.lower(),v))
    elif col.lower()=='watermark':
        new=input("Enter new watermark:")
        cursor2.execute("update note set watermark='{}' where
country='{}' and denomination={}".format(new,c.lower(),v))
    else:
        print("No such field found.")
global A
A=False
mycon2.commit()
#####
def addc(c): #adding coin details
    print("\nNOTE: If any field is unknown then please enter 'na'.")
    n=input("Enter your name: ")
    v=int(input("Denomination:"))
    cu2=str(input("Currency name: "))
    y=str(input("Year of issue: "))
    f=input("Design on front side:")
    b=input("Design on back side: ")
    e=input("Material: ")
    print("Dimensions↓")
    l=input("Diameter: ")
    s=input("Thickness: ")
    ti=str(time.strftime('%d-%m-%y'))

mycon=my.connect(host='localhost',user='root',passwd='tejas123',database='numis')
cursor=mycon.cursor()
q3="insert into coin
values('{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}')".format(c,v,y,f
,b,e,l,s,n,ti)

```

```

cursor.execute(q3)
mycon.commit()
print("Thanks for helping!! :D")
mycon.close()
q5="insert into currency values ('{}','{}')".format(c,cu2)
cursor2.execute(q5)
mycon2.commit()
print(131* "=")
#####
#####
##### adding note
def addn(c):
    details
    print("\nNOTE: If any field is unknown then please enter 'na'.")
    n=input("Enter your name: ")
    v=int(input("Denomination: "))
    cu2=input("Currency name: ")
    y=str(input("Year of issue: "))
    f=input("Design on front side: ")
    b=input("Design on back side: ")
    e=input("Material: ")
    print("Dimensions↓")
    l=input("Length: ")
    s=input("Breadth: ")
    w=input("Watermark: ")
    ti=str(time.strftime('%d-%m-%y'))

mycon=my.connect(host='localhost',user='root',passwd='tejas123',database='numis')
cursor=mycon.cursor()
q4="insert into note
values ('{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}')".format(c,
v,y,f,b,e,l,s,w,n,ti)
cursor.execute(q4)
mycon.commit()
print("Thanks for helping!! :D")
mycon.close()
q5="insert into currency values ('{}','{}')".format(c,cu2)
cursor2.execute(q5)
mycon2.commit()
print(131* "=")
#####
#####
##### country(c):

```

```

mycon=my.connect(host='localhost',user='root',passwd='tejas123',database='numis')

cursor=mycon.cursor()
t=input("Do you want to know about coin or note?(c/n):")
if t.lower()=='c':
    q="select denomination from coin where country='{}' order by denomination".format(c,)
    cursor.execute(q)
    data=cursor.fetchall()
    if len(data)!=0:
        print("The available coins of",c.title(),"are:")
#If type= coin
    for i in range(0,len(data)):
        print(data[i][0])
    print("NOTE: If you want to add information about a coin which is not mentioned above,\nthen enter that denomination.")
    v=int(input("Choose the denomination: "))
    print("\n")
    q2="select * from coin where country='{}' and denomination={}".format(c.lower(),v)
    cursor.execute(q2)
    data1=cursor.fetchone()
    if data1!=None:
        print(7*"\t"," DESCRIPTION")
        for i in range(len(data1)):
            if i==6:
                print("Dimensions\t")
                print(C[i],str(data1[i]).capitalize())
            else:
                print("Sorry! We don't have any information about this country. :(")
    a=input("Would you like to help us know about it? (y/n) :")

    if a.lower()=='n':
        print("No problem! :)")
    elif a.lower()!="y":
        print("Invalid reply! >:( ")
    else:
        addn(c)

else:
    print("Sorry! We don't have any information about this country. :(")

```

```

a=input("Would you like to help us know about it? (y/n) :")

if a.lower() == "n":
    print("No problem! :)")
elif a.lower() != "y":
    print("Invalid reply! >:( ")
else:
    addn(c)

elif t.lower() == 'n':
    q="select denomination from note where country='{}' order by
denomination".format(c,)
    cursor.execute(q)
    data=cursor.fetchall()
    if len(data)!=0:
        print("The available notes of",c.title(),"are:")
# if type= note
    for i in range(len(data)):
        print(data[i][0])
    print("NOTE: If you want to add information about a note
which is not mentioned above,\nthen enter that denomination.")
    v=int(input("Choose the denomination:"))
    print("\n")
    q2="select * from note where country='{}' and
denomination={}".format(c,v)
    cursor.execute(q2)
    data1=cursor.fetchone()
    if data1!=None:
        print(7*"\\t"," DESCRIPTION")
        for i in range(len(data1)):
            if i==6:
                print("Dimensions↓")
            print(N[i],str(data1[i]).capitalize())
    else:
        print("Sorry! We don't have any information about this
country. :(")
    a=input("Would you like to help us know about it?
(y/n) :")

    if a.lower() == "n":
        print("No problem! :)")
    elif a.lower() != "y":
        print(" Invalid reply! >:( ")

```

```

        else:
            addn(c)

    else:
        print("Sorry! We don't have any information about this
country. :(")
        a=input("Would you like to help us know about it? (y/n):")
        if a.lower() == "n":
            print("No problem! :)")
        elif a.lower() != "y":
            print("Invalid reply! >:(")
        else:
            addn(c)

    else:
        print("Invalid reply! >:(")
        mycon.close()
        print(131* "=")

# Function to view items in the shopping cart
def view_items(cursor,u):
    cursor.execute("SELECT * FROM shopping_cart where
username='{}'".format(u))
    items = cursor.fetchall()
    if items:
        print("Shopping Cart Items:")
        for i in range(0,len(items)):
            print("#",items[i][0])
    else:
        print("Shopping cart is empty.")

# Function to add an item to the shopping cart
def add_item(cursor, item_id,u):
    try:
        q="insert into shopping_cart values({},{})".format(item_id,u)
        cursor.execute(q)
        print("Item added successfully.")
    except Exception as err:
        print("Error adding item :(")
    mycon.commit()

# Function to delete an item from the shopping cart
def delete_item(cursor, item_id,u):
    try:
        cursor.execute("DELETE FROM shopping_cart WHERE id = {} and
username='{}'".format(item_id,u))

```



```

Mode\n",5*"\\t","5. User Login",3*"\\t","6. New User?\n",5*"\\t","7.
Exit\n")
o=int(input("Enter your choice:"))

if o==1:
#DISPLAY information
    c=str(input("Enter the name of country:"))

mycon2=my.connect(host='localhost',user='root',passwd='tejas123',database='numis')
    cursor2=mycon2.cursor()
    cursor2.execute("select cu from currency where
country='%s'"%(c.lower().strip(),))
    cu=cursor2.fetchone()
    if cu==None:
        print("Sorry! We don't have any information about this
country. :(")
        a=input("Would you like to help us know about it? (y/n):")
        if a.lower()=='n':
            print("No problem! :) ")
        elif a.lower()!='y':
            print("Invalid reply! >:(")
        else:
            ty=input("Do you want to add information about coin or
note? (c/n):")
            if ty.lower()=='c':
                addc(c)
            elif ty.lower()=='n':
                addn(c)
            else:
                print("Invalid Reply! >:(")

    else:
        print("The currency of",c.title(),"is:",cu[0].title())
        country(c.lower().strip())

elif o==2:
#ADD information
    c=str(input("Enter the name of country:"))
    ty=input("Do you want to add information about coin or note?
(c/n):")
    cursor2=mycon2.cursor()
    if ty=='c':
        addc(c)

```

```

    elif ty=='n':
        addn(c)
    else:
        print("Invalid reply! >:(")

    elif o==3:
#UPDATE information
        ty=input("Do you want to update details of coin or note?
(c/n) :")
        c=input("Enter the name of country:")
        v=int(input("Enter the denomination:"))
        cursor2=mycon2.cursor()
        if ty.lower()=='c':
            cursor2.execute("select * from coin where country='{}' and
denomination={}".format(c,v))
            data=cursor2.fetchone()
            if data!=None:
                print(7*"\t","CURRENT DETAILS")
                for i in range(len(data)):
                    if i==6:
                        print("Dimensions↓")
                        print(C[i],str(data[i]).capitalize())
                col=input("Which field do you want to update?: ")
                A=True
                update_c(col,c,v)
                if A==True:
                    print("Details successfully updated.")

            else:
                print("No such information set found!")

        elif ty.lower()=='n':
            cursor2.execute("select * from note where country='{}' and
denomination={}".format(c,v))
            data=cursor2.fetchone()
            if data!=None:
                print(7*"\t","CURRENT DETAILS")
                for i in range(len(data)):
                    if i==6:
                        print("Dimensions↓")
                        print(N[i],str(data[i]).capitalize())
                col=input("Which field do you want to update?: ")
                A=True
                update_n(col,c,v)

```

```

        if A==True:
            print("Details successfully updated.")

        else:
            print("No such information set found!")

    else:
        print("Invalid Reply >:(")

elif o==4:
#ADMIN MODE

    try:
        p = getpass.getpass()
    except Exception as error:
        print('There was some error: ', error)
    else:
        if p=="tob":
            print("Hello Sir,")
            print("Database changed.")
            q=""
            while(q!="exit"):
                q=input("")
                if q.lower()=="exit":
                    print("Exited the database.")
                    break
                cursor2=mycon2.cursor()
                cursor2.execute(q)
                data=cursor2.fetchall()
                if len(data)!=0:
                    for i in range(0,len(data)):
                        print(data[i])
                else:
                    print("Operation performed.")
                mycon2.commit()

        else:
            print("Incorrect Password >:(")
            z+=1
            if z>=2:
                print("There have been more than 1 failed login
attempts. Closing the system.")
                time.sleep(0.3)
                print(".",end="")
                time.sleep(0.3)

```

```

        print(".",end="")
        time.sleep(0.3)
        print(".",end="")
        time.sleep(3)
        break
    print(131*"=")

elif o==5:
    u=input("Enter the username: ")
    try:
        p = getpass.getpass()
    except Exception as error:
        print('There was some error: ', error)
    else:

mycon2=my.connect(host='localhost',user='root',passwd='tejas123',database='numis')
    cursor2=mycon2.cursor()
    cursor2.execute("select pwd from login where
username='%s'"%(u.lower().strip(),))
    pwd=cursor2.fetchone()
    if pwd==None:
        print("No account exists with that username.")
    else:
        if pwd[0]==p:
            print("Login Successful.")

mycon=my.connect(host='localhost',user='root',passwd='tejas123',database='numis')
    cursor=mycon.cursor()
    while True:
        print("\nShopping Cart Menu:")
        print("1. View Items")
        print("2. Add Item")
        print("3. Delete Item")
        print("4. Buy Now")
        print("5. Logout")

    choice = input("Enter your choice: ")

    if choice == "1":
        view_items(cursor,u)
    elif choice == "2":
        item_id = input("Enter the ID of the item
to add: ")

```

```

        add_item(cursor, item_id,u)
    elif choice == "3":
        item_id = input("Enter the ID of the item
to delete: ")
        delete_item(cursor, item_id,u)
    elif choice == "4":
        buy_now(cursor)
    elif choice == "5":
        break
    else:
        print("Invalid choice. Please enter a valid
option.")

cursor.close()
mycon.close()
print("Logged Out.")

else:
    print("Incorrect Password >:( ")
    z+=1
    if z>=2:
        print("There have been more than 1 failed login
attempts. Closing the system.")
        time.sleep(0.3)
        print(".",end="")
        time.sleep(0.3)
        print(".",end="")
        time.sleep(0.3)
        print(".",end="")
        time.sleep(3)
        break
    print(131* "=")

elif o==6:
    u=input("Enter the username: ")
    p=input("Set a password: ")

mycon2=my.connect(host='localhost',user='root',passwd='tejas123',database='numis')
cursor2=mycon2.cursor()
cursor2.execute("insert into login
values ('%s','%s') "%(u.lower().strip(),p.lower().strip()))
mycon2.commit()

```

```

    elif o==7:
#EXIT
        print("Thanks for visiting :)")
        print(131* "=")
        time.sleep(0.2)
        print(".",end="")
        time.sleep(0.2)
        print(".",end="")
        time.sleep(0.2)
        print(".")
        time.sleep(2)
        break
    else:
        print("Invalid choice! >:()")
        print(131* "=")
        print(131* "=")
        choice=input("Do you want to continue? (y/n) :")
        if choice.lower()=='n':
            print("Thanks for visiting :)")
        if choice.lower() not in ('y','n'):
            print("Invalid reply! >:()")
        mycon2.close()
print(131* "=")
print("Redirecting to Frontend")
time.sleep(0.4)
print(".",end="")
time.sleep(0.4)
print(".",end="")
time.sleep(0.4)
print(".")
time.sleep(0.4)
webbrowser.open('https://multiverseweb.github.io/login/')

```

Database

```
mysql> use numis;
Database changed
mysql> show tables;
+-----+
| Tables_in_numis |
+-----+
| coin           |
| currency       |
| login          |
| note           |
+-----+
4 rows in set (0.00 sec)

mysql> desc coin;
+-----+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| country    | varchar(200) | YES  |     | NULL    |       |
| denomination | int        | YES  |     | NULL    |       |
| year       | varchar(200) | YES  |     | NULL    |       |
| front      | varchar(200) | YES  |     | NULL    |       |
| back       | varchar(200) | YES  |     | NULL    |       |
| material    | varchar(200) | YES  |     | NULL    |       |
| diameter    | varchar(200) | YES  |     | NULL    |       |
| thickness   | varchar(200) | YES  |     | NULL    |       |
| name        | varchar(200) | YES  |     | NULL    |       |
| time        | varchar(200) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0.03 sec)

mysql> desc note;
+-----+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| country    | varchar(200) | YES  |     | NULL    |       |
| denomination | int        | YES  |     | NULL    |       |
| year       | varchar(200) | YES  |     | NULL    |       |
| front      | varchar(200) | YES  |     | NULL    |       |
| back       | varchar(200) | YES  |     | NULL    |       |
| material    | varchar(200) | YES  |     | NULL    |       |
| length      | varchar(200) | YES  |     | NULL    |       |
| breadth     | varchar(200) | YES  |     | NULL    |       |
| watermark   | varchar(200) | YES  |     | NULL    |       |
| name        | varchar(200) | YES  |     | NULL    |       |
| time        | varchar(200) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)
```

```

mysql> desc currency;
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| country | varchar(200) | YES |     | NULL    |       |
| cu       | varchar(200) | YES |     | NULL    |       |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> desc login;
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| username | varchar(50) | YES |     | NULL    |       |
| pwd      | varchar(50) | YES |     | NULL    |       |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from login;
+-----+-----+
| username | pwd      |
+-----+-----+
| tejas    | numismatist |
| ojas     | ojas      |
+-----+-----+
2 rows in set (0.01 sec)

mysql> select * from currency;
+-----+-----+
| country | cu        |
+-----+-----+
| india   | indian rupee |
| pakistan | pakistani rupee |
| india   | Indian rupee |
| pakistan | Pakistani Rupee |
| France  | Francs    |
| Canada  | Canadian Dollar |
| america | american dollar |
| america | american dollar |
| Bhutan  | Ngultrum  |
| America | american dollar |
| switzerland | Swiss Francs |
| india   | Indian Rupee |
| indonesia | indonesian rupiah |
| singapore | singapore dollar |
| yugoslavia | Yugoslavian Dinara |
| china   | jiao      |
| sri lanka | sri lankan rupee |
| nepal   | nepali rupee |
| denmark | denmark krone |
+-----+-----+

```

34 rows in set (0.01 sec)

Olive branch and oak branch with denomination.				
nickel				
1.71mm Muskan 20-03-22	26.5mm			
indonesia 1000 2010	denomination below the			
arms of indonesia. "BANK INDONESIA 1000 RUPIAH"				
The Angklung, a musical instrument and the Gedung State, a public building in bandung in west java.				
24.15mm 1.6mm tejas 21-05-22	Nickel plated steel			
singapore 1 2013 to 2018	na			
muskan 21-05-22				
nepal 1 2007 to 2009	Year of issue, Mount			
Everest and "Sagarmatha" (name of Everest in Nepali) as the inscription.				
"???????".				
Nepal Map with mountains and value in English above and in Nepali below.				
"NEPAL Re-1 ????? ? ???????".	brass plated steel			
20mm 1.8mm tejas 25-05-22				
denmark 1 1973 to 1989	Portrait of margrethe ii			
facing right. "margrethe ii danmarks dronning b ? b".				
Value, crowned royal coat of arms. "1974 1 krone".				
copper-nickel				
1.6mm tejas 26-05-22	25.5mm			
japan 10 1959 to 1989	Phoenix Hall (H??d?			
Temple, By?d?-in) in center with Country name on top and value below. "? ? ?				
? ??" i.e. "State of Japan 10 Yen".				
Value and denomination within wreath. "10 ??????" Translation: Year 49 of				
Sh?wa.	Bronze (95% Copper,			
4-3% Zinc, 1-2% Tin)	23.5mm 1.5mm Tejas			
30-05-22				
germany 2 2002 to 2022	An oak twig, symbolizing			
stability, surrounded by the twelve stars of Europe. "A 2006"				
A globe, next to the face value, shows Europe in relation to Africa and Asia.				
"2 EURO CENT LL"	Copper plated steel			
18.75mm 1.67mm tejas 30-05-22				
russia 10 1991 (Soviet Union) Kremlin Tower and Dome.				
"???????????????? ???? · ???? ·" Translation: Government Bank USSR.				
Value flanked by sprigs above date. "10 ?????? M 1991" Translation: 10				
Kopecks.	copper clad steel			
17.9mm 1.25mm tejas 30-05-22				
hong kong 2 1993 to 2019	Bauhinia or Hong Kong			
orchid flower (Binomial Name: Bauhinia blakeana). "? ? HONG KONG" Translation:				
Hong Kong.				
The value in numerals above the date, with the denomination in Latin below.				
"? ? 2 1994 TWO DOLLARS" Translation: Two Dollars. Copper nickel				
28mm 2.03mm Tejas 02-06-22				
switzerland 2 1968 to 2022	Full-length figure of			
helvetia, standing, wearing toga, with spear and shield with coat of arms of				
switzerland. Stars representing the 23 cantons of the Swiss confederation.				

"HELVETIA A . BOVY INC?.". | Wreath formed of oak and alpine rose branches tied with a ribbon below. "2 Fr. 1988 B".

| copper(75%) and nickel (25) | 27.40mm |
2.15mm | muskan | 02-06-22 |

15 rows in set (0.02 sec)

mysql> select * from note;

country	denomination	year	front

back

material	length		
breadth	watermark	name	time

| India | 2000 | 2016 to 2019 | Portrait of Mahatma Gandhi with
denomination and Reserve Bank logo. "Reserve Bank of India"

Mangalyaan with language panel and denomination. "TWO THOUSAND RUPEES"

Paper	166mm		
66mm	Mahatma Gandhi and denomination.	Tejas	11-03-22
india	500	2016 to 2021	Mahatma Gandhi with denomination, reserve bank logo, national emblem of India "RESERVE BANK OF INDIA"

Red Fort, Delhi with flag of India. "FIVE HUNDRED RUPEES"

Paper	149mm		
65mm	Mahatma Gandhi and denomination	Tejas	12-03-22
pakistan	10	2006 to 2021	Quaid-e-azam muhammad ali jinnah

Khyber pass gameway and state bank seal

| paper | 115mm
| 65mm | Muhammad ali jinnah and denomination | Ojas | 12-03-22 |
| india | 200 | 2017 to 2020 | Mahatma Gandhi with national emblem and reserve bank logo. "RESERVE BANK OF INDIA"

Sanchi stupa with language panel. "TWO HUNDRED RUPEES"

| paper | 146mm
| 67mm | Mahatma Gandhi and denomination | ojas | 12-03-22 |
| india | 100 | 2018 to 2021 | Mahatma Gandhi with reserve bank seal, national emblem of india, denomination. "RESERVE BANK OF INDIA 100"
| A picture of Rani Ki Vav. "RESERVE BANK OF INDIA ONE HUNDRED RUPEES 100 RANI KI VAV"
| Paper | 142mm
| 65mm | Mahatma Gandhi with Denomination. | Tejas | 15-03-22 |
| india | 20 | 2019 to 2021 | Mahatma Gandhi

Ellora Caves

| Paper | 129mm
| 63mm | Mahatma Gandhi and denomination | Tejas | 17-03-22 |
| Bhutan | 1 | 1981 to 1985 | Royal emblem between facing dragons at center, serial number at upper left and right.
"?????????????????????", "??1"
| Simtokha Dzong palace.

| "ROYAL GOVERNMENT OF BHUTAN", "?ONE NGULTRUM?????????????1" | 155mm
| 67mm | no watermark | Tejas | 12-04-22 |
| America | 1 | 1963 to 2017 | Portrait of George Washington at center.

Treasurers Signature and Federal Reserve Bank Seal at left.

| Secretaries Signature and Treasury Seal at right. | "FEDERAL RESERVE NOTE, THE UNITED STATES OF AMERICA, THIS NOTE IS LEGAL TENDER, FOR ALL DEBTS, PUBLIC AND PRIVATE, WASHINGTON, ONE DOLLAR" | 67mm | no watermark
| Ojas | 12-04-22 |
| india | 10 | na | na | na

| Paper | na
| na | mahatma gandhi and denomination | Tejas | 09-05-22 |
| yugoslavia | 500 | 1970 to 1986 | Statue of Nikola Tesla in Niagara Falls State Park on Goat Island in New York "???????? ???? ?????????? 500
?????A"
"500 ?ET CTOT?HA ?????A SOCIALISTI?KA FEDERATIVNA REPUBLIKA JUGOSLAVIJA"

| Paper | 155mm
| 74mm | no watermark | Tejas | 23-05-22 |
| china | 1 | 1962 | Brown on multicolour underprint.
workers at left. "?????? ??"

| 1
in decorative design. Arms at right. "ZHONGGUO RENMIN YINHANG 1 1962"
| paper | 104mm
| 50mm | no watermark | Tejas | 24-05-22 |
| sri lanka | 20 | 2010 to 2020 | Port of Colombo at center, butterfly lower left, owl at right.

Ves Netuma dancer and Geta Bera drummer. "????? ???? ?? ?????? ?????? ?????? ??????
????? CENTRAL BANK OF SRI LANKA ?????? ?????? ?????? TWENTY
RUPEES" | paper

128mm

| 67mm | owl and number 20 | tejas | 25-05-22 |
| eritrea | 1 |

|

|

| | tejas | 01-06-22 |
| oman | 100 | 1416 (1995) | Deep olive-green, dark green-blue
and purple on multicolor underprint. Sultan qaboos bin sa-id at right. faslajs
irrigation system at center. central bank of oman one hundred baisa ??? |
Verreaux eagle and white oryx at center. CENTRAL BANK OF OMAN ONE HUNDRED
BAISA 100

| Paper | 122mm
| 64mm | Sultan Qaboos bin Said | nimish | 25-06-22 |
| lebanon | 10 | 1964 to 1986 | Ruins of Anjar

Pigeons Rock, in Beirut BANQUE DU LIBAN 10 LIVRES DIX LIVRES

| Paper | 145mm
| 75mm | Na |
| Ojas | 26-06-22 |
| myanmar | 10 | 1996 to 1997 | Purple colour, Chinte
"??????????????????????" ?? 10 ??????" Translation: Central bank of
myanmar 10 10 Ten Kyats.
| A karaweik (royal regalia boat) "Central bank of myanmar 10 ten kyats".
| paper | 130mm
| 60mm | Chinthe head and value in Burmese (??) | Tejas | 26-06-22 |
| tatarstan | 100 | 1991 | Tatar flag and castle

blank

| paper | 140mm
| 66mm | lozenges | Tejas | 01-09-23 |
+-----+-----+-----+
-----+-----+-----+
-----+-----+-----+
-----+-----+-----+
-----+-----+-----+
-----+-----+-----+
17 rows in set (0.01 sec)

Output: Implementation Result

```
...
=====
★ WELCOME TO NUMISMATIC GALLERY! ★

Numismatic Gallery is an interface to facilitate the posting and fetching of information of currency which is part of our
collection. This program will allow viewers of the collection to look for the currency they are interested in by entering the
country it belongs to. In case the collector does not have any currency of that country, it will allow the user to add
information about it from his side by filling details in fields like year, description, country, denomination and composition.

-Tejas, Ojas & Bhavya
=====
...
    ★ MENU ★
1. Display details      2. Add details
3. Update details       4. Admin Mode
5. User Login           6. New User?
7. Exit

Enter your choice:1
Enter the name of country:india
The currency of India is: Indian Rupee
Do you want to know about coin or note?(c/n):n
The available notes of India are:
10
20
100
200
500
2000
NOTE: If you want to add information about a note which is not mentioned above,
then enter that denomination.
Choose the denomination:2000

DESCRIPTION
Country: India
Denomination: 2000
Year: 2016 to 2019
Frontside: Portrait of mahatma gandhi with denomination and reserve bank logo. "reserve bank of india"
Backside: Mangalyaan with language panel and denomination. "two thousand rupees"
Material: Paper
Dimensions↓
Length: 166mm
Breadth: 66mm
Watermark: Mahatma gandhi and denomination.

© Tejas
Lastupdated on: 11-03-22
=====
=====
Do you want to continue? (y/n):y
    ★ MENU ★
1. Display details      2. Add details
3. Update details       4. Admin Mode
5. User Login           6. New User?
7. Exit

Enter your choice:2
Enter the name of country:australia
Do you want to add information about coin or note? (c/n):n

NOTE: If any field is unknown then please enter 'na'.
Enter your name: Tejas
Denomination: 5
Currency name: Australian Dollars
Year of issue: na
Design on front side: na
Design on back side: na
Material: na
Dimensions↓
Length: na
Breadth: na
Watermark: na
Thanks for helping!! :D
=====
```

Do you want to continue? (y/n):y

★ MENU ★

- | | |
|--------------------|----------------|
| 1. Display details | 2. Add details |
| 3. Update details | 4. Admin Mode |
| 5. User Login | 6. New User? |
| 7. Exit | |

Enter your choice:1

Enter the name of country:australia

The currency of Australia is: Australian Dollars

Do you want to know about coin or note?(c/n):n

The available notes of Australia are:

5

NOTE: If you want to add information about a note which is not mentioned above,
then enter that denomination.

Choose the denomination:5

DESCRIPTION

Country: Australia
Denomination: 5
Year: Na
Frontside: Na
Backside: Na
Material: Na
Dimensions↓
Length: Na
Breadth: Na
Watermark: Na

© Tejas
Lastupdated on: 15-10-23

=====

Do you want to continue? (y/n):y

★ MENU ★

- | | |
|--------------------|----------------|
| 1. Display details | 2. Add details |
| 3. Update details | 4. Admin Mode |
| 5. User Login | 6. New User? |
| 7. Exit | |

Enter your choice:3

Do you want to update details of coin or note? (c/n):n

Enter the name of country:australia

Enter the denomination:5

CURRENT DETAILS

Country: Australia
Denomination: 5
Year: Na
Frontside: Na
Backside: Na
Material: Na
Dimensions↓
Length: Na
Breadth: Na
Watermark: Na

© Tejas
Lastupdated on: 15-10-23

Which field do you want to update?: year

Enter new year:2012

No such information set found!

=====

Do you want to continue? (y/n):y

★ MENU ★

- | | |
|--------------------|----------------|
| 1. Display details | 2. Add details |
| 3. Update details | 4. Admin Mode |
| 5. User Login | 6. New User? |
| 7. Exit | |

Enter your choice:4

Warning: Password input may be echoed.

Password: bot

Incorrect Password >:(

=====

=====

Do you want to continue? (y/n):y

★ MENU ★

- | | |
|--------------------|----------------|
| 1. Display details | 2. Add details |
|--------------------|----------------|

- 3. Update details
- 4. Admin Mode
- 5. User Login
- 6. New User?
- 7. Exit

```

Enter your choice:4
Warning: Password input may be echoed.
Password: tob
Hello Sir,
Database changed.
show tables
('coin',)
('currency',)
('login',)
('note',)
('shopping_cart',)
select * from currency
('india', 'indian rupee')
('pakistan', 'pakistani rupee')
('india', 'Indian rupee')
('pakistan', 'Pakistani Rupee')
('France', 'Francs')
('Canada', 'Canadian Dollar')
('america', 'american dollar')
('america', 'american dollar')
('Bhutan', 'Ngultrum')
('America', 'american dollar')
('switzerland', 'Swiss Francs')
('india', 'Indian Rupee')
('indonesia', 'indonesian rupiah')
('singapore', 'singapore dollar')
('yugoslavia', 'Yugoslavian Dinara')
('china', 'jiao')
('sri lanka', 'sri lankan rupee')
('nepal', 'nepali rupee')
('denmark', 'denmark krone')
('japan', 'yen')
('germany', 'Euro')
('russia', 'russian ruble')
('eritrea', 'nakfa')
('Hong Kong', 'Hong Kong Dollar')
('lebanon', 'lebanon livres')
('switzerland', 'swiss franc')
('switzerland', 'swiss franc')
('oman', 'Omani baiza')
('lebanon', 'Livres (pounds)')
('myanmar', 'Myanmar Kyats')
('tejas', 'tejas')
('cyprus', 'cypriot pound')
('tatarstan', 'Ruble')
('uk', 'na')
('pakistan', 'na')
('ok', 'na')
('australia', 'Australian Dollars')
exit
Exited the database.
=====
```

Do you want to continue? (y/n):y

★ MENU ★

- 1. Display details
- 2. Add details
- 3. Update details
- 4. Admin Mode
- 5. User Login
- 6. New User?
- 7. Exit

```

Enter your choice:6
Enter the username: tejas
Set a password: admin
=====
```

Do you want to continue? (y/n):y

★ MENU ★

- 1. Display details
- 2. Add details
- 3. Update details
- 4. Admin Mode
- 5. User Login
- 6. New User?
- 7. Exit

```

Enter your choice:5
Enter the username: tejas
Warning: Password input may be echoed.
```

```

Password: admin
Incorrect Password >:( 
There have been more than 1 failed login attempts. Closing the system.
=====
Redirecting to Frontend
...
= RESTART: C:\Users\iamte\Desktop\TejasCodes\python\tejas_ngms.py
...
=====
★ WELCOME TO NUMISMATIC GALLERY! ★

Numismatic Gallery is an interface to facilitate the posting and fetching of information of currency which is part of our
collection. This program will allow viewers of the collection to look for the currency they are interested in by entering the
country it belongs to. In case the collector does not have any currency of that country, it will allow the user to add
information about it from his side by filling details in fields like year, description, country, denomination and composition.

-Tejas, Ojas & Bhavya
=====
★ MENU ★
...
1. Display details      2. Add details
3. Update details       4. Admin Mode
5. User Login           6. New User?
7. Exit

Enter your choice:5
Enter the username: ojas
Warning: Password input may be echoed.
Password: ojas
Login Successful.

Shopping Cart Menu:
1. View Items
2. Add Item
3. Delete Item
4. Buy Now
5. Logout
Enter your choice: 2
Enter the ID of the item to add: 260205
Item added successfully.

Shopping Cart Menu:
1. View Items
2. Add Item
3. Delete Item
4. Buy Now
5. Logout
Enter your choice: 1
Shopping Cart Items:
# 987
# 7492
# 1287
# 260205

Shopping Cart Menu:
1. View Items
2. Add Item
3. Delete Item
4. Buy Now
5. Logout
Enter your choice: 3
Enter the ID of the item to delete: 260205
Item deleted successfully.

Shopping Cart Menu:
1. View Items
2. Add Item
3. Delete Item
4. Buy Now
5. Logout
Enter your choice: 1
Shopping Cart Items:
# 987
# 7492
# 1287
Shopping Cart Menu:
1. View Items

```

2. Add Item
3. Delete Item
4. Buy Now
5. Logout
Enter your choice: 4
Items purchased successfully.

Shopping Cart Menu:

1. View Items
2. Add Item
3. Delete Item
4. Buy Now
5. Logout

Enter your choice: 5

Logged Out.

=====

Do you want to continue? (y/n):y

★ MENU ★

- | | |
|--------------------|----------------|
| 1. Display details | 2. Add details |
| 3. Update details | 4. Admin Mode |
| 5. User Login | 6. New User? |
| 7. Exit | |

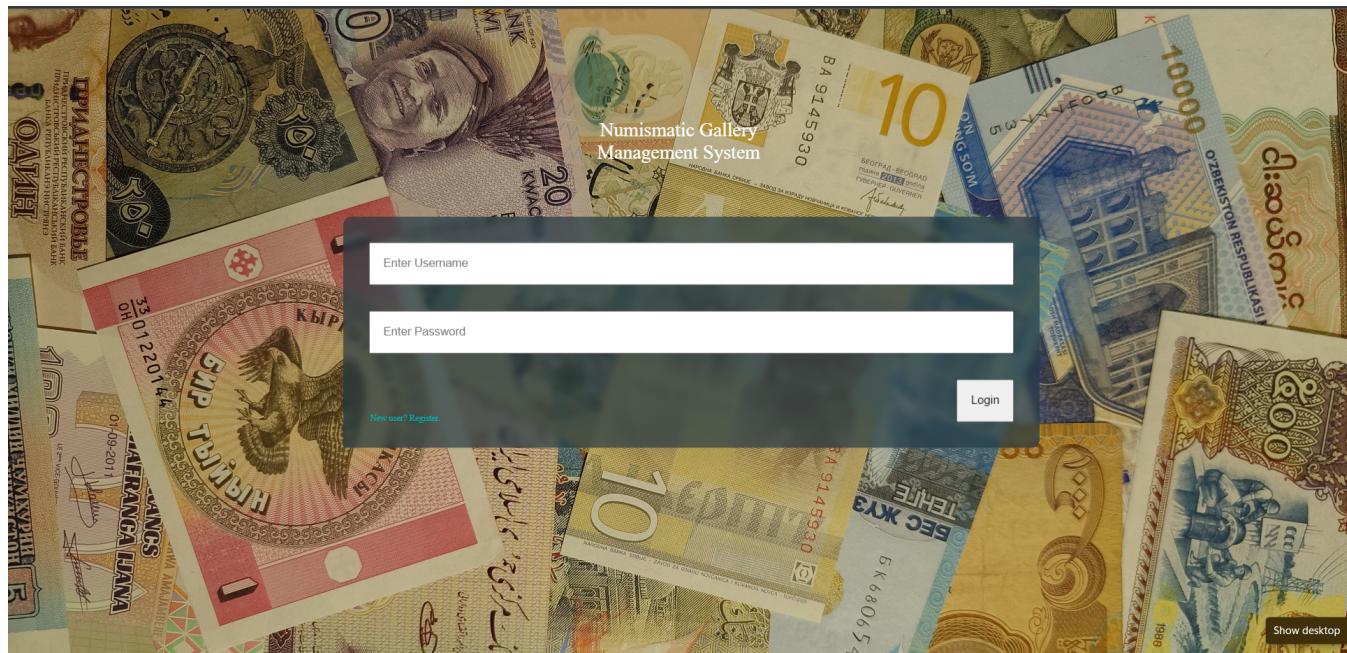
Enter your choice:7

Thanks for visiting :)

=====
...
=====

Redirecting to Frontend

...



Advantages of the Collection Management System:

1. User-Friendly Interface:

- Advantage: The system provides a straightforward and user-friendly interface, making it easy for users to manage their collections and make purchases.
- Reasoning: The simplicity of the interface enhances user experience, especially for individuals who may not be technically inclined.

2. CRUD Operations:

- Advantage: Incorporates fundamental CRUD operations, enabling users to easily create accounts, add items to their collection, and manage their shopping cart.
- Reasoning: CRUD operations are essential for any data management system, providing users with the ability to interact with their data in a meaningful way.

3. File Handling:

- Advantage: Utilizes file handling for persistent data storage, ensuring that user data is saved between program sessions.
- Reasoning: File handling is a practical solution for small-scale applications, providing a simple and accessible means of storing and retrieving data.

4. Admin Mode:

- Advantage: Includes an admin mode with password protection, allowing administrators to view the contents of all user files.
- Reasoning: Admin functionality is crucial for system maintenance and monitoring, providing oversight and control over user data.

5. Scalability:

- Advantage: The project serves as a foundation that can be extended for the development of more comprehensive collection management systems.
- Reasoning: The modular structure and simplicity of the program make it adaptable for future enhancements and additional features.

Disadvantages of the Collection Management System:

1. Limited Security Features:

- Disadvantage: The project lacks advanced security features such as data encryption, which could pose a security risk for sensitive user information.
- Reasoning: For a production environment, it is essential to implement robust security measures to protect user data from unauthorized access.

2. Basic Error Handling:

- Disadvantage: The program has limited error handling, which may result in unexpected behavior or crashes in response to invalid user inputs.

- Reasoning: Robust error handling is crucial for providing a more resilient and user-friendly application.

3. Dependence on File Handling:

- Disadvantage: While file handling is a simple solution, it may not be suitable for large-scale applications, where a relational database management system (RDBMS) like MySQL might be more appropriate.

- Reasoning: RDBMS offers better data organization, consistency, and scalability compared to file handling.

4. Simplistic Admin Mode:

- Disadvantage: The admin mode is basic and primarily focused on viewing user files. It lacks more advanced administrative functionalities.

- Reasoning: In a real-world scenario, an admin mode might require more features such as user management, analytics, and system configuration.

5. Limited Data Attributes:

- Disadvantage: The project only includes basic attributes for items (name, type, value). Additional attributes such as images, descriptions, or purchase history are not considered.

- Reasoning: Expanding the data model could provide users with a more comprehensive and informative view of their collections.

In summary, while the Collection Management System Project (Python and MySQL) provides a solid foundation, addressing certain limitations such as security, error handling, and database management could significantly enhance its functionality and suitability for real-world deployment.

Applications of Numismatic Gallery Management System

The Collection Management System has various applications, and it can be utilized in different scenarios where individuals or organizations need to organize, track, and manage collections of items. Here are some potential applications:

Personal Collections:

Application: Individuals who collect items such as coins, banknotes, stamps, trading cards, or any other collectibles can use the system to keep a digital record of their collection. It helps in easy tracking, management, and sharing of their collections.

Museum Collections:

Application: Museums and cultural institutions can use a more sophisticated version of this system to manage their artifact collections. It helps in cataloging items, tracking their provenance, and managing exhibitions.

Libraries and Archives:

Application: Libraries and archives can adapt the system to manage collections of rare books, manuscripts, or historical documents. It assists in tracking loans, managing inventory, and preserving valuable items.

Art Galleries:

Application: Art galleries can utilize a similar system to manage their art collections. It can include details about each artwork, artist information, exhibition history, and pricing.

Educational Institutions:

Application: Educational institutions with programs related to archaeology, history, or cultural studies can use the system for educational purposes. Students can manage simulated collections as part of their coursework.

Auction Houses:

Application: Auction houses can benefit from a collection management system to catalog and organize items that are up for auction. It helps in tracking bids, managing transactions, and providing a detailed history of items.

Hobbyist Communities:

Application: Online communities centered around specific hobbies or collectibles can integrate a collection management system. It facilitates interaction among members, sharing of collection details, and trading or selling items within the community.

In summary, the Collection Management System has versatile applications across various domains, providing a structured and organized way to manage and track collections of diverse items. Its adaptability makes it a valuable tool for both individual enthusiasts and institutional entities.

Conclusion

In summary, the Collection Management System, developed using Python and MySQL, seamlessly integrated with a dedicated website frontend, stands as a sophisticated solution for enthusiasts managing their coin and banknote collections. The project's strength lies in the combination of Python's flexibility for backend logic and MySQL's prowess as a relational database, fostering efficient data management.

The MySQL database serves as a robust backend, ensuring the structured storage of user account details and collection items. This relational model facilitates quick and organised retrieval, updating, and deletion of data, contributing to a smooth and responsive user experience.

The frontend, embedded within a user-friendly website, significantly elevates the accessibility and usability of the system. The graphical interface enhances user interaction, offering features such as responsive design, dynamic updates, and intuitive navigation. This not only simplifies the user experience but also opens avenues for customization, branding, and extended user engagement.

Moreover, Python's adaptability augurs well for future expansions. The system can seamlessly incorporate advanced features, such as fortified user authentication methods, detailed item descriptions, and sophisticated search functionalities. The combination of a powerful backend, a user-centric frontend, and Python's extensibility establishes a foundation that accommodates the evolving needs of collectors.

In essence, the Collection Management System, with its Python-MySQL synergy and a dedicated website frontend, transcends the basic requirements of a collection management platform. It not only caters to the fundamental functionalities but also lays the groundwork for a scalable, customizable, and immersive user experience in the ever-evolving world of collection management.

References

Books:

Title: "Computer Science With Python Class 12"

- Author: Sumita Arora
- Publisher: Dhanpat Rai & Co.
- Year: 2021

Online Resources:

Installation and setting up:

[W3Schools.com](https://www.w3schools.com/python/python_mysql_getstarted.asp)

Other References:

Frontend: <https://en.numista.com/>
