Project

Virtual Art Gallery

Name:Devaki Akash

Artist class:

```
class Artist:
    def __init__(self, artistid, name,artworkid,biography,birthdate,nationality,website,ContactInformation):
        self.__artistid = artistid
        self.__name = name
        self.__artworkid=artworkid
        self.__biography=biography
        self.__birthdate=birthdate
        self.__nationality=nationality
        self.__website=website
        self.__ContactInformation=ContactInformation

    @property
    def getartist_id(self):
        return self.__artistid
    @property
    def getname(self):
        return self.__name
    @property
    def getartworkid(self):
        return self.__artworkid
    @property
    def getbiography(self):
        return self.__biography
    @property
    def getbirthdate(self):
        return self.__birthdate
```

```
    @property
    def getnationality(self):
        return self.__nationality
    @property
    def getwebsite(self):
        return self.__website
    @property
    def getContactInformation(self):
        return self.__ContactInformation

    @getartist_id.setter
    def setartist_id(self,artistid):
       self.__artistid=artistid
    @getname.setter
    def setname(self,name):
        self.__name=name
    @getartworkid.setter
    def setartworkid(self,artworkid):
        self.__artworkid=artworkid
    @getbiography.setter
    def setbiography(self,biography):
        self.__biography=biography
    @getbirthdate.setter
    def setbirthdate(self,birthdate):
        self.__birthdate=birthdate
```

```
    @getnationality.setter
    def setnationality(self,nationality):
        self.__nationality=nationality
    @getwebsite.setter
    def setwebsite(self,website):
        self.__website=website
    @getContactInformation.setter
    def setContactInformation(self,ContactInformation):
         self.__ContactInformation=ContactInformation
```

Artwork class:

```
class Artwork:
    def __int__(self,artworkid,title,description,creationdate,medium,artisid,imageurl):
        self.__artworkid = artworkid
        self.__title = title
        self.__description = description
        self.__creationdate = creationdate
        self.__medium = medium
        self.__image_url = imageurl

    @property
    def getartworkid(self):
        return self.__artwork_id
    @property
    def gettitle(self):
        return self.__title
    @property
    def getdescription(self):
        return self.__description
    @property
    def getcreationdate(self):
        return self.__creation_date
    @property
    def getmedium(self):
        return self.__medium
    @property
    def getimageurl(self):
        return self.__image_url
```

```
    @getartworkid.setter
    def setartworkid(self, artworkid):
        self.__artworkid = artworkid
    @gettitle.setter
    def settitle(self, title):
        self.__title = title
    @getdescription.setter
    def setdescription(self, description):
        self.__description= description
    @getcreationdate.setter
    def setcreationdate(self, creationdate):
        self.__creationdate = creationdate
    @getmedium.setter
    def setmedium(self, medium):
        self.__medium = medium
    @getimageurl.setter
    def setimageurl(self, imageurl):
        self.__imageurl = imageurl
```

User Class:

```
class user:
    def __int__(self,userid,username,password,email,firstname,lastname,dateofbirth,profilepicture):
        self.__userid=userid
        self.__username=username
        self.__password=password
        self.__email=email
        self.__firstname=firstname
        self.__last=lastname
        self.dateofbirth=dateofbirth
        self.profilepicture=profilepicture

    @property
    def getuserid(self):
        return self.__userid
    @property
    def getusername(self):
        return self.__username
    @property
    def getpassword(self):
        return self.__password
    @property
    def getemail(self):
        return self.__email
    @property
    def getfirstname(self):
        return self.__firstname
    @property
    def getlastname(self):
        return self.__lastname
```

```
    @property
    def getdateofbirth(self):
        return self.__dateofbirth
    @property
    def getprofilepicture(self):
        return self.__profilepicture

    @getuserid.setter
    def setuserid(self, userid):
        self.__usertid = userid
    @getusername.setter
    def setusername(self, username):
        self.__username = username
    @getpassword.setter
    def setpassword(self, password):
        self.__password = password
    @getemail.setter
    def setemail(self, email):
        self.__email = email
    @getfirstname.setter
    def setfirstname(self, firstname):
        self.__firstname = firstname
    @getlastname.setter
    def setlastname(self, lastname):
        self.__lastname = lastname
    @getdateofbirth.setter
    def setdateofbirth(self, dateofbirth):
        self.__dateofbirth = dateofbirth
```

```
    @getprofilepicture.setter
    def setprofilepicture(self, profilepicture):
        self.__profilepicture = profilepicture
```

Gallery class:

```
class gallery:
    def __int__(self,galleryid,name,description,location,curator,openinghours):
        self.__galleryid=galleryid
        self.__name=name
        self.__description=description
        self.__location=location
        self.__curator=curator
        self.__openinghours=openinghours

    @property
    def getgalleryid(self):
        return self.__galleryid
    @property
    def getname(self):
        return self.__name
    @property
    def getdescription(self):
        return self.__description
    @property
    def getlocation(self):
        return self.__location
    @property
    def getcurator(self):
        return self.__curator
    @property
    def getopeninghours(self):
        return self.__openinghours
```

```
    @getgalleryid.setter
    def setgalleryid(self, galleryid):
        self.__galleryid = galleryid
    @getname.setter
    def setname(self, name):
        self.__name= name
    @getdescription.setter
    def setdescription(self, description):
        self.__description = description
    @getlocation.setter
    def setlocation(self, location):
        self.__location = location
    @getcurator.setter
    def setcurator(self, curator):
        self.__curator = curator
    @getopeninghours.setter
    def setopeninghours(self, openinghours):
        self.__openinghours = openinghours
```

Abstract Method:

```python
from abc import ABC, abstractmethod
class IVirtualArtGallery(ABC):

    # Artwork Management
    @abstractmethod
    def add_artwork(self):
        pass

    @abstractmethod
    def update_artwork(self, artwork):
        pass

    @abstractmethod
    def remove_artwork(self, artwork_id):
        pass

    @abstractmethod
    def get_artwork_by_id(self, artwork_id):
        pass

    @abstractmethod
    def search_artworks(self, keyword):
        pass

    @abstractmethod
    def add_artwork_to_favorite(self, user_id, artwork_id):
        pass
```

```python
    @abstractmethod
    def remove_artwork_from_favorite(self, user_id, artwork_id):
        pass

    @abstractmethod
    def get_user_favorite_artworks(self, user_id):
        pass
```

Exceptions:

```
class ArtWorkNotFoundException(Exception) :
    pass
class UserNotFoundException(Exception):
    pass
class GalleryNotFoundException(Exception):
    pass
```

Database connection:

```
import mysql.connector
from mysql.connector import Error
from configparser import ConfigParser


class dbutil:
    def __init__(self,host,user,password,port,database):
        self.connection=mysql.connector.connect(
            host = 'localhost',
            user = 'root',
            password = 'root',
            port = '3306',
            database = 'virtualartgallery'
        )
        self.cursor=self.connection.cursor()
```

Create Artist:

```
from myExceptions import *
class VirtualArtGalleryImplementation(IVirtualArtGallery,ArtWorkNotFoundException,UserNotFoundException):
    flag=0

    #test1=MyTestCase()
    def createartist(self):
        artistid = self.get_unique_artistid()
        artworkid = self.get_unique_artworkid()
        name = input("enter the artist name")
        biography = input("enter artist biography")
        birthdate = input("enter the bithdate of artist")
        nationality = input("enter the nationality of artist")
        website = input("enter the url of website")
        contactinformation = input("enter the gmail of artist")
        ca = {
            'artistid': artistid,
            'artworkid': artworkid,
            'name': name,
            'biography': biography,
            'birthdate': birthdate,
            'nationality': nationality,
            'website': website,
            'contactinformation': contactinformation
        }
```

```
query = "insert into artist values(%s,%s,%s,%s,%s,%s,%s,%s)"
values = (ca['artistid'], ca['artworkid'], ca['name'], ca['biography'], ca['birthdate'], ca['nationality'],
          ca['website'], ca['contactinformation'])
cur.execute(query, values)
cur.fetchall()
con.commit()
```

```
def all_artist(self):
    query="select * from artist"
    cur.execute(query)
    return cur.fetchall()
def get_unique_artistid(self):
    return len(self.all_artist())+1
```

Create Artwork:

```
def add_artwork(self):
    flag=0
    try:
        artworkid = self.get_unique_artworkid()
        artistid = self.get_unique_artistid()
        title= input("enter the title of the artwork")
        description = input("enter the description")
        creationdate = input("enter the creation date")
        medium = input("enter the medium")
        artwork = {
            'artworkid': artworkid,
            'artistid': artistid,
            'title':title,
            'description': description,
            'creationdate': creationdate,
            'medium': medium
        }
        self.createartist()
        v.add_artwork_to_db(artwork)
        con.commit()
        flag=1
    finally:
        con.commit()
    return flag
```

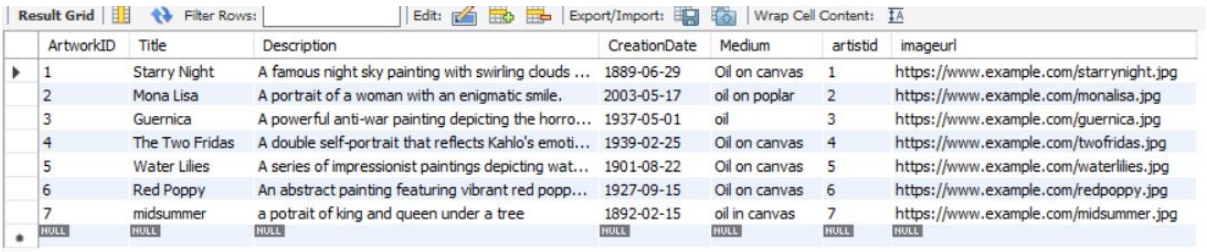
```python
def add_artwork_to_db(self,artwork):

    query = "insert into artwork values(%s,%s,%s,%s,%s,%s)"
    values = (
    artwork['artworkid'], artwork['title'], artwork['description'], artwork['creationdate'],artwork['medium'],
    artwork['artistid'])
    cur.execute(query, values)
    cur.fetchall()
    con.commit()
```

```python
    def all_artwork(self):
        query="select * from artwork"
        cur.execute(query)
        return cur.fetchall()
    def get_unique_artworkid(self):
        return len(self.all_artwork())+1
```

Output:

```
"D:\Hexaware\Foundation Training\tool code\Assignments\python\project-1\
choose the options from given below
1.add_artwork
2.update_Artwork
3.remove artwork
4.get artwork by id
5.search artwork
6.add artwork to favorite
7.remove artwork from favorites
8.get user favorite artwork
9.exit
enter the option you have choosen1
enter the artist namejames
enter artist biographyscottish painter
enter the bithdate of artist1859-11-20
enter the nationality of artistscottish
enter the url of websitehttps://collections.royalscottishacademy.org/
enter the gmail of artistjames@gmail.com
enter the title of the artworkmidsummer
enter the descriptiona potrait of king and queen under a tree
enter the creation date1892-02-15
enter the mediumoil in canvas
enter the urlhttps://www.example.com/midsummer.jpg
artwork added successfully into the database
```

artwork_table:



ArtworkID	Title	Description	CreationDate	Medium	artistid	imageurl
1	Starry Night	A famous night sky painting with swirling clouds ...	1889-06-29	Oil on canvas	1	https://www.example.com/starrynight.jpg
2	Mona Lisa	A portrait of a woman with an enigmatic smile.	2003-05-17	oil on poplar	2	https://www.example.com/monalisa.jpg
3	Guernica	A powerful anti-war painting depicting the horro...	1937-05-01	oil	3	https://www.example.com/guernica.jpg
4	The Two Fridas	A double self-portrait that reflects Kahlo's emoti...	1939-02-25	Oil on canvas	4	https://www.example.com/twofridas.jpg
5	Water Lilies	A series of impressionist paintings depicting wat...	1901-08-22	Oil on canvas	5	https://www.example.com/waterlilies.jpg
6	Red Poppy	An abstract painting featuring vibrant red popp...	1927-09-15	Oil on canvas	6	https://www.example.com/redpoppy.jpg
7	midsummer	a potrait of king and queen under a tree	1892-02-15	oil in canvas	7	https://www.example.com/midsummer.jpg
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Artist Table:

ArtistID	artworkid	Name	Biography	BirthDate	Nationality	Website	ContactInformation
1	1	Vincent van Gogh	Dutch post-impressionist painter.	1853-03-30	Dutch	https://www.vangoghgallery.com/	info@vangogh.com
2	2	Leonardo da Vinci	Italian polymath of the Renaissance.	1977-04-15	Italian	https://www.leonardodavinci.net/	contact@leonardo.com
3	3	Pablo Picasso	Spanish painter and sculptor.	1900-10-25	Spanish	https://www.picasso.com/	picasso@artworld.com
4	4	Frida Kahlo	Mexican painter known for her self-portraits.	1907-07-06	Mexican	https://www.fridakahlo.org/	frida.kahlo@email.com
5	5	Claude Monet	French impressionist painter.	1870-11-14	French	https://www.claudemonetgallery.org/	monet@impressionism.com
6	6	Georgia O'Keeffe	American modernist artist.	1887-11-15	American	https://www.georgiaokeeffe.net/	info@okeeffeart.com
7	7	james	scottish painter	1859-11-20	scottish	https://collections.royalscottishacademy.org/	james@gmail.com
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Update-artwork:

```
def update_artwork(self,artworkid):
    flag = 0
    try:
        query="select * from artwork where artworkid= %s"
        cur.execute(query,(artworkid,))
        output=cur.fetchall()
        if not output:
            raise ArtWorkNotFoundException(f"Error: artwork {artworkid} not found")
            return False
        print("select the below options to update:")
        print("1.title")
        print("2.description")
        print("3.creationdate")
        print("4.medium")
        choice = input("enter the option you want to change")
        if choice == "1":
            title = input("enter the title")
            query = "update artwork set title=%s where artworkid= %s"
            cur.execute(query, (title,artworkid,))
            cur.fetchone()
            con.commit()
            print("Artwork updated successfully")
            return True
```

```
elif choice == "2":
    description = input("enter the description")
    query = "update artwork set description=%s where artworkid= %s"
    cur.execute(query, (description,artworkid,))
    cur.fetchone()
    con.commit()
    print("Artwork updated successfully")
    return True
elif choice == "3":
    creationdate = input("enter the creationdate")
    query = "update artwork set creationdate=%s where artworkid= %s"
    cur.execute(query, (creationdate,artworkid,))
    cur.fetchone()
    con.commit()
    print("Artwork updated successfully")
    return True
elif choice == "4":
    medium = input("enter the medium")
    query = "update artwork set medium=%s where artworkid= %s"
    cur.execute(query, (medium,artworkid,))
    cur.fetchone()
    con.commit()
    print("Artwork updated successfully")
    return True
```

```
    else:
        print("invalid choose from above option")
    query = "select * from artwork where artworkid=%s"
    cur.execute(query,(artworkid,))
    output = cur.fetchone

    con.commit()
except ArtWorkNotFoundException as e:
    print(f"Error: {e}")
return False
```

Output:

```
choose the options from given below
1.add_artwork
2.update_Artwork
3.remove artwork
4.get artwork by id
5.search artwork
6.add artwork to favorite
7.remove artwork from favorites
8.get user favorite artwork
9.exit
enter the option you have choosen2
enter the artworkid2
select the below options to update:
1.title
2.description
3.creationdate
4.medium
enter the option you want to change4
enter the mediumoil in canvas
Artwork updated successfully
```

Database:

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

ArtworkID	Title	Description	CreationDate	Medium	artistid	imageurl
1	Starry Night	A famous night sky painting with swirling clouds ...	1889-06-29	Oil on canvas	1	https://www.example.com/starrynight.jpg
2	Mona Lisa	A portrait of a woman with an enigmatic smile.	2003-05-17	oil in canvas	2	https://www.example.com/monalisa.jpg
3	Guernica	A powerful anti-war painting depicting the horro...	1937-05-01	Oil on canvas	3	https://www.example.com/guernica.jpg
4	The Two Fridas	A double self-portrait that reflects Kahlo's emoti...	1939-02-25	Oil on canvas	4	https://www.example.com/twofridas.jpg
6	Red Poppy	An abstract painting featuring vibrant red popp...	1927-09-15	Oil on canvas	6	https://www.example.com/redpoppy.jpg
7	midsummer	a potrait of king and queen under a tree	1892-02-15	oil in poplar	7	https://www.example.com/midsummer.jpg
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
"D:\Hexaware\Foundation Training\tool code\Assignments\python\project\venv\Scripts\python.exe" "D:/Hexaware/Foundation Training/tool code/Assignments/python/project/main.py"
(1, 'Starry Night', 'A famous night sky painting with swirling clouds and bright stars.', datetime.date(1889, 6, 29), 'Oil on canvas', 1, 'https://www.example.com/starrynight.jpg
(2, 'Mona Lisa', 'A portrait of a woman with an enigmatic smile.', datetime.date(2003, 5, 17), 'oil in canvas', 2, 'https://www.example.com/monalisa.jpg')
(3, 'Guernica', 'A powerful anti-war painting depicting the horrors of the bombing of Guernica.', datetime.date(1937, 5, 1), 'Oil on canvas', 3, 'https://www.example.com/guernica
(4, 'The Two Fridas', "A double self-portrait that reflects Kahlo's emotions after her divorce.", datetime.date(1939, 2, 25), 'Oil on canvas', 4, 'https://www.example.com/twofrid
(6, 'Red Poppy', 'An abstract painting featuring vibrant red poppy flowers.', datetime.date(1927, 9, 15), 'Oil on canvas', 6, 'https://www.example.com/redpoppy.jpg')
(7, 'midsummer', 'a potrait of king and queen under a tree', datetime.date(1892, 2, 15), 'oil in poplar', 7, 'https://www.example.com/midsummer.jpg')
```

Remove_artwork:

```python
def remove_artwork(self, artworkid):
    flag=0
    try:
        query = "select * from artwork where artworkid=%s"
        cur.execute(query, (artworkid,))
        output = cur.fetchone()
        query = "delete from artwork where artworkid=%s"
        cur.execute(query, (artworkid,))
        if not output:
            raise ArtWorkNotFoundException(f"Error: artwork {artworkid} not found")
        for i in output:
            print(i)
        print(f"deleted the {artworkid} from database successfully")
        con.commit()
        flag=1
    except ArtWorkNotFoundException as e:
        print(f"Error: {e}")
```

Output:

```
1.add_artwork
2.update_Artwork
3.remove artwork
4.get artwork by id
5.search artwork
6.add artwork to favorite
7.remove artwork from favorites
8.get user favorite artwork
9.exit
enter the option you have choosen3
enter the artwork_id3
3
Guernica
A powerful anti-war painting depicting the horrors of the bombing of Guernica.
1937-05-01
Oil on canvas
None
https://www.example.com/guernica.jpg
deleted the 3 from database successfully
```

Datebase:

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

ArtworkID	Title	Description	CreationDate	Medium	artistid	imageurl
1	Starry Night	A famous night sky painting with swirling clouds ...	1889-06-29	oil	1	https://www.example.com/starrynight.jpg
2	Mona Lisa	A portrait of a woman with an enigmatic smile.	2003-05-17	oil in canvas	2	https://www.example.com/monalisa.jpg
4	The Two Fridas	A double self-portrait that reflects Kahlo's emoti...	1939-02-25	Oil on canvas	4	https://www.example.com/twofridas.jpg
6	Red Poppy	An abstract painting featuring vibrant red popp...	1927-09-15	Oil on canvas	6	https://www.example.com/redpoppy.jpg
7	midsummer	a potrait of king and queen under a tree	1892-02-15	oil in poplar	7	https://www.example.com/midsummer.jpg
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Get Artwork By Id:

```
def get_artwork_by_id(self, artworkid):
    try:
        query = "select * from artwork where artworkid=%s"
        cur.execute(query, (artworkid,))
        output=cur.fetchall()
        print(cur.fetchall())
        con.commit()
        if not output:
            raise ArtWorkNotFoundException(f"Error:artworkid {artworkid} not found ")
        for i in output:
            print(i)
    except ArtWorkNotFoundException as e:
        print(f"{e}")
```

Output:

```
choose the options from given below
1.add_artwork
2.update_Artwork
3.remove artwork
4.get artwork by id
5.search artwork
6.add artwork to favorite
7.remove artwork from favorites
8.get user favorite artwork
9.exit
enter the option you have choosen4
enter the artwork_id2
[(2, 'Mona Lisa', 'A portrait of a woman with an enigmatic smile.', datetime.date(2003, 5, 17), 'oil on poplar', 2, 'https://www.example.com/monalisa.jpg')]
choose the options from given below
```

Search artwork :

```python
def search_artworks(self, title):
    flag=0
    try:
        query = "select * from artwork where title=%s"
        cur.execute(query, (title,))
        output=cur.fetchall()
        if not output:
            raise ArtWorkNotFoundException(f"Error: artwork {title} not found")
        for i in output:
            print(i)

        flag=1
        return True

    except ArtWorkNotFoundException as e:
        print(f"{e}")
    return flag
```

Output:

```
1.add_artwork
2.update_Artwork
3.remove artwork
4.get artwork by id
5.search artwork
6.add artwork to favorite
7.remove artwork from favorites
8.get user favorite artwork
9.exit
enter the option you have choosen5
enter the name red poppy
(6, 'Red Poppy', 'An abstract painting featuring vibrant red poppy flowers.', datetime.date(1927, 9, 15), 'Oil on canvas', 6, 'https://www.example.com/redpoppy.jpg')
```

Add user favorite artwork:

```
def add_artwork_to_favorite(self, userid, artworkid):
    try:
        query="select * from UserFavoriteArtwork where userid=%s"
        cur.execute(query)
        output=cur.fetchall()
        if not output:
            raise UserNotFoundException(f"Error {userid} userid not found")
        self.get_all_users()
        self.userid=userid
        self.artworkid=artworkid
        self.get_all_artwork()

        fav = {
            'userid': userid,
            'artworkid': artworkid
        }
        query = "insert into UserFavoriteArtwork values(%s,%s)"
        values = (fav['userid'], fav['artworkid'])
        cur.execute(query, values)
        output = cur.fetchall()
        for i in output:
            print(i)
        print(output, " successfully inserted into database")
        con.commit()
    except UserNotFoundException as e:
        print(f"Error: {e}")
```

Output:

```
choose the options from given below
1.add_artwork
2.update_Artwork
3.remove artwork
4.get artwork by id
5.search artwork
6.add artwork to favorite
7.remove artwork from favorites
8.get user favorite artwork
9.exit
enter the option you have choosen6
enter the userid1
enter the artworkid3
```

Database:

	userid	artworkid
▶	1	2
	1	4
	2	3
	2	4
	3	1
	3	2
	4	5
	5	2
	5	4
	5	5
	6	1
	1	3

Remove from user favotite:

```
def remove_artwork_from_favorite(self, userid, artworkid):
    try:
        query="select * from UserFavoriteArtwork where userid=%s"
        cur.execute(query)
        output=cur.fetchall()
        if not output:
            raise UserNotFoundException(f"Error: {userid} userid not found")
        self.get_all_users()
        self.userid = userid
        self.artworkid = artworkid
        self.get_all_artwork()

        query = "delete from UserFavoriteArtwork where userid=%s and artworkid=%s"
        cur.execute(query, (userid, artworkid,))
        print("successfully deleted from the database")
        con.commit()
    except UserNotFoundException as e:
        print(f"{e}")
```

Output:

```
choose the options from given below
1.add_artwork
2.update_Artwork
3.remove artwork
4.get artwork by id
5.search artwork
6.add artwork to favorite
7.remove artwork from favorites
8.get user favorite artwork
9.exit
enter the option you have choosen7
enter the userid1
enter the artwork_id3
```

Database:

userid	artworkid
1	2
1	4
2	3
2	4
3	1
3	2
4	5
5	2
5	4
5	5
6	1

Get User favorite:

```
def get_user_favorite_artworks(self, userid):
    try:

        self.get_all_users()
        userid = input("enter the userid")
        query = "select artworkid from UserFavoriteArtwork where userid=%s "
        cur.execute(query, (userid,))
        output = cur.fetchall()
        if not output:
            raise UserNotFoundException(f"Error: {userid} userid not found")

        for i in output:
            print(i)
        con.commit()
    except UserNotFoundException as e:
        print(f" {e}")
```

Output:

```
choose the options from given below
1.add_artwork
2.update_Artwork
3.remove artwork
4.get artwork by id
5.search artwork
6.add artwork to favorite
7.remove artwork from favorites
8.get user favorite artwork
9.exit
enter the option you have choosen8
(1, 'artlover123', 'password123', 'artlover@example.com', 'John', 'Doe', datetime.date(1990, 5, 20), 'profile1.jpg')
(2, 'paintingFanatic', 'securepass', 'fanatic@email.com', 'Jane', 'Smith', datetime.date(1985, 12, 10), 'profile2.jpg')
(3, 'creativeSoul', 'myp@ssw0rd', 'creativesoul@mail.com', 'Alice', 'Johnson', datetime.date(1992, 8, 15), 'profile3.jpg')
(4, 'artExplorer', 'exploreart', 'explorer@gmail.com', 'Bob', 'Williams', datetime.date(1988, 3, 25), 'profile4.jpg')
(5, 'museumGoer', 'visitmuseums', 'museumgoer@example.com', 'Charlie', 'Davis', datetime.date(1995, 6, 3), 'profile5.jpg')
(6, 'colorEnthusiast', 'colorfulpass', 'colorful@email.com', 'Eva', 'Clark', datetime.date(1982, 11, 28), 'profile6.jpg')
enter the userid2
(3,)
(4,)
```

Exit:

```
choose the options from given below
1.add_artwork
2.update_Artwork
3.remove artwork
4.get artwork by id
5.search artwork
6.add artwork to favorite
7.remove artwork from favorites
8.get user favorite artwork
9.exit
enter the option you have choosen9
exiting the system
 Thank you
```

Create Gallery:

```
def create_new_gallery(self):
    flag=0
    try:
        galleryid=self.get_unique_gallery_id();
        name=input("enter the gallery name")
        description=input("enter the description")
        location=input("enter the location")
        curator=input("enter the artistid or curator")
        openinghours=input("enter the opening hours")
        new={
            'galleryid':galleryid,
            'name':name,
            'description':description,
            'location':location,
            'curator':curator,
            'openinghours':openinghours
        }
        query="insert into gallery values(%s,%s,%s,%s,%s,%s)"
        values=(new['galleryid'],new['name'],new['description'],new['location'],new['curator'],new['openinghours'])
        cur.execute(query,values)
        output=cur.fetchall()
        for i in output:
            print(i)
    finally:
        con.commit()
```

```python
def get_all_gallaries(self):
    query="select * from gallery"
    cur.execute(query)
    return cur.fetchall()

def get_unique_gallery_id(self):
    return len(self.get_all_gallaries())+1
```

Output:

```
"D:\Hexaware\Foundation Training\tool code\Assignments\p
enter the gallery nameArt house
enter the descriptioncity's artistic legacy
enter the locationkolkata
enter the artistid or curator 13
enter the opening hoursMon-Thur 9am-4pm sat 9am -6pm

Process finished with exit code 0
```

DataBase:

galleryid	Name	Description	Location	curator	OpeningHours
7	Art house	city's artistic legacy	kolkata	13	Mon-Thur 9am-4pm sat 9am -6pm
101	Artistic	A contemporary art gallery	123 Main Street, Cityville	11	Mon-Fri: 10 am - 6 pm, Sat-Sun: 12 pm - 4 pm
102	Classic Impressions	Specializing in classic art pieces	456 Oak Avenue, Townsville	12	Tue-Sat: 9 am - 5 pm
103	Modern Expression	Showcasing modern and abstract art	789 Elm Street, Artburg	13	Wed-Mon: 11 am - 7 pm
104	Cultural Hub Gallery	Diverse collection representing various cultures	101 Pine Road, Cultura	14	Thu-Sun: 1 pm - 8 pm
105	Nature's Canvas Gallery	Focused on nature-inspired artworks	202 Maple Lane, Green City	15	Fri-Sun: 10 am - 5 pm
106	Innovative Art Space	Promoting innovative and experimental art forms	303 Cedar Street, Creatopia	16	Mon-Wed: 12 pm - 6 pm, Sat: 10 am - 4 pm
NULL	NULL	NULL	NULL	NULL	NULL

Update Gallery:

```
def update_gallery(self):
    self.get_all_gallaries()
    galleryid=input("enter the gallery id")
    try:
        query="select * from gallery where galleryid=%s"
        cur.execute(query,(galleryid,))
        output=cur.fetchall()
        if not output:
            raise GalleryNotFoundException(f"Error: artwork {galleryid} not found")
        print("select the below options to update:")
        print("1.Name")
        print("2.description")
        print("3.location")
        print("4.curator")
        print("5.openinghours")
        choice = input("enter the option you want to change")
        if choice == "1":
            name = input("enter the Name")
            query = "update gallery set name=%s where galleryid= %s"
            cur.execute(query, (name,galleryid,))
            cur.fetchone()
            con.commit()
            print("successfully updated gallery name")
        elif choice == "2":
            description = input("enter the description")
            query = "update gallery set description=%s where galleryid= %s"
            cur.execute(query, (description,galleryid,))
            cur.fetchone()
            con.commit()
```

```
    elif choice == "3":
        location = input("enter the location")
        query = "update gallery set location=%s where galleryid= %s"
        cur.execute(query, (location,galleryid,))
        cur.fetchone()
        con.commit()
        print("successfully updated gallery location")
    elif choice == "4":
        curator = input("enter the curator")
        query = "update gallery set curator=%s where galleryid= %s"
        cur.execute(query, (curator,galleryid,))
        cur.fetchone()
        con.commit()
        print("successfully updated gallery curator")
    elif choice =="5":
        openinghours = input("enter the openinghours")
        query = "update gallery set openinghours=%s where galleryid= %s"
        cur.execute(query, (openinghours, galleryid,))
        cur.fetchone()
        con.commit()
        print("successfully updated gallery openinghours")
    else:
        print("invalid choose from above option")
    query = "select * from gallery where galleryid=%s"
    cur.execute(query,(galleryid,))
    output=cur.fetchone
```

```
except GalleryNotFoundException as e:
    print(f"Error: {e}")
```

Output:

```
enter the gallery id101
select the below options to update:
1.Name
2.description
3.location
4.curator
5.openinghours
enter the option you want to change1
enter the NameArtistic
successfully updated gallery name

Process finished with exit code 0
```

Database:

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

galleryid	Name	Description	Location	curator	OpeningHours
101	Artistic	A contemporary art gallery	123 Main Street, Cityville	11	Mon-Fri: 10 am - 6 pm, Sat-Sun: 12 pm - 4 pm
102	Classic Impressions	Specializing in classic art pieces	456 Oak Avenue, Townsville	12	Tue-Sat: 9 am - 5 pm
103	Modern Expression	Showcasing modern and abstract art	789 Elm Street, Artburg	13	Wed-Mon: 11 am - 7 pm
104	Cultural Hub Gallery	Diverse collection representing various cultures	101 Pine Road, Cultura	14	Thu-Sun: 1 pm - 8 pm
105	Nature's Canvas Gallery	Focused on nature-inspired artworks	202 Maple Lane, Green City	15	Fri-Sun: 10 am - 5 pm
106	Innovative Art Space	Promoting innovative and experimental art forms	303 Cedar Street, Creatopia	16	Mon-Wed: 12 pm - 6 pm, Sat: 10 am - 4 pm
NULL	NULL	NULL	NULL	NULL	NULL

Remove gallery:

```
def remove_gallery(self):
    self.get_all_gallaries()
    galleryid=input("enter the galleryid to remove")
    flag=0
    try:
        query = "delete from gallery where galleryid=%s"
        cur.execute(query, (galleryid,))
        output = cur.fetchall()
        if not output:
            print(f"Error: artwork {galleryid} not found")
        for i in output:
            print(i)
        print(f"deleted the {galleryid} from database successfully")
        con.commit()
        flag=1
    finally:
        con.commit()
```

Output:

```
"D:\Hexaware\Foundation Training\tool code\As
enter the galleryid to remove7
deleted the 7 from database successfully

Process finished with exit code 0
```

Database:

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

galleryid	Name	Description	Location	curator	OpeningHours
101	Artistic	A contemporary art gallery	123 Main Street, Cityville	11	Mon-Fri: 10 am - 6 pm, Sat-Sun: 12 pm - 4 pm
102	Classic Impressions	Specializing in classic art pieces	456 Oak Avenue, Townsville	12	Tue-Sat: 9 am - 5 pm
103	Modern Expression	Showcasing modern and abstract art	789 Elm Street, Artburg	13	Wed-Mon: 11 am - 7 pm
104	Cultural Hub Gallery	Diverse collection representing various cultures	101 Pine Road, Cultura	14	Thu-Sun: 1 pm - 8 pm
105	Nature's Canvas Gallery	Focused on nature-inspired artworks	202 Maple Lane, Green City	15	Fri-Sun: 10 am - 5 pm
106	Innovative Art Space	Promoting innovative and experimental art forms	303 Cedar Street, Creatopia	16	Mon-Wed: 12 pm - 6 pm, Sat: 10 am - 4 pm
NULL	NULL	NULL	NULL	NULL	NULL

Search gallery:

```
def search_gallery(self ):
    flag=0
    self.get_all_gallaries()
    galleryid=input("enter the galleryid")
    try:
        query = "select * from gallery where galleryid=%s"
        cur.execute(query, (galleryid,))
        output=cur.fetchall()
        if not output:
            print(f"Error: artwork {galleryid} not found")
        for i in output:
            print(i)

        flag=1
    finally:
        con.commit()
```

```
enter the galleryid101
(101, 'Artistic Haven', 'A contemporary art gallery', '123 Main Street, Cityville', 11, 'Mon-Fri: 10 am - 6 pm, Sat-Sun: 12 pm - 4 pm')

Process finished with exit code 0
```

Art Gallery App:

```
from dao.implementation import VirtualArtGalleryImplementation
from dao import test_mytesting
from dao.test_mytesting import MyTestCase

crime=VirtualArtGalleryImplementation()
def main():
    crime=VirtualArtGalleryImplementation()
    test=MyTestCase()
    while True:
        print("choose the options from given below")
        print("1.add_artwork")
        print("2.update_Artwork")
        print("3.remove artwork")
        print("4.get artwork by id")
        print("5.search artwork")
        print("6.add artwork to favorite")
        print("7.remove artwork from favorites")
        print("8.get user favorite artwork")
        print("9.exit")
        choice=input("enter the option you have choosen")
        if choice=="1":
            crime.add_artwork()

        elif choice=="2":
            artworkid=input("enter the artworkid")
            crime.update_artwork(artworkid)
```

```
elif choice=="3":
    artworkid = input("enter the artwork_id")
    crime.remove_artwork(artworkid)
elif choice=="4":
    artworkid=input("enter the artwork_id")
    crime.get_artwork_by_id(artworkid)
elif choice=="5":
    artworkid=input("enter the artwork_id")
    crime.search_artworks(artworkid)
    test.test_searching_artwork()
elif choice=="6":
    userid = input("enter the userid")
    artworkid = input("enter the artworkid")

    crime.add_artwork_to_favorite(userid,artworkid)
elif choice=="7":
    userid = input("enter the userid")
    artworkid = input("enter the artwork_id")
    crime.remove_artwork_from_favorite(userid,artworkid)
elif choice=="8":
    userid = ("enter the userid")
    crime.get_user_favorite_artworks(userid)
elif choice=="9":
    print("exiting the system \n Thank you")
    break
else:
    print("invalid option choose from above given options")
```

Artwork Management:

```
import unittest

from dao.implementation import VirtualArtGalleryImplementation


class MyTestCase(unittest.TestCase):


    virtual=VirtualArtGalleryImplementation()

    def test_searching_artwork(self):
        flag=2
        self.assertEqual(self.virtual.search_artworks(flag),True)
    def test_removing_artwork(self):
        flag=3
        self.assertEqual(self.virtual.remove_artwork(flag),True)
    def test_update_artwork(self):

        self.assertEqual(self.virtual.update_artwork.flag,True)
    def test_add_artwork(self):

        self.assertEqual(self.virtual.add_artwork.flag, True)




if __name__ == '__main__':
    unittest.main()
```

Output:

```
test_mytesting.py::MyTestCase::test_removing_artwork
test_mytesting.py::MyTestCase::test_searching_artwork
test_mytesting.py::MyTestCase::test_update_artwork

========================= 2 failed, 2 passed in 0.48s =========================

Process finished with exit code 1
```

Gallery Management:

```python
import unittest
from dao.implementation import VirtualArtGalleryImplementation,flag


class MyTestCase(unittest.TestCase):
    virtual = VirtualArtGalleryImplementation()
    def test_search_gallery(self):
        galleryid=101
        self.assertEqual(self.virtual.search_gallery(galleryid),True)
    def test_removing_gallery(self):
        gallery_id=101
        self.assertEqual(self.virtual.remove_gallery(gallery_id),True)

    def test_update_gallery(self):
        self.assertEqual(self.virtual.update_gallery.flag, True)

    def test_add_gallery(self):
        self.assertEqual(self.virtual.create_new_gallery.flag, True)

if __name__ == '__main__':
    unittest.main()
```

Output:

```
test_gallerytesting.py::MyTestCase::test_removing_gallery
test_gallerytesting.py::MyTestCase::test_search_gallery
test_gallerytesting.py::MyTestCase::test_update_gallery

========================== 2 failed, 2 passed in 0.46s ==========================

Process finished with exit code 1
```