ASSIGNMENT 2

STUDENT INFORMATION SYSTEMS

AKILESH K

Task 1 & 2

```
class Student:
    def __init__(self, student_id, first_name, last_name, dob, email, phone_number):
        self.student_id = student_id
        self.first_name = first_name
        self.last_name = last_name
        self.dob = dob
        self.email = email
        self.phone_number = phone_number

class Course:
    def __init__(self, course_id, course_name, course_code, instructor_name):
        self.course_id = course_id
        self.course_name = course_name
        self.course_code = course_code
        self.instructor_name = instructor_name

class Enrollment:
    def __init__(self, enrollment_id, student_id, course_id, enrollment_date):
        self.enrollment_id = enrollment_id
        self.student_id = student_id
        self.course_id = course_id
        self.enrollment_date = enrollment_date

class Teacher:
    def __init__(self, teacher_id, first_name, last_name, email):
        self.teacher_id = teacher_id
```

```
        self.first_name = first_name
        self.last_name = last_name
        self.email = email

class Payment:
    def __init__(self, payment_id, student_id, amount, payment_date):
        self.payment_id = payment_id
        self.student_id = student_id
        self.amount = amount
        self.payment_date = payment_date
```

TASK 3

```
    def enroll_in_course(self, course):
        self.enrolled_courses.append(course)

    def update_student_info(self, first_name, last_name, date_of_birth, email, phone_number):
        self.first_name = first_name
        self.last_name = last_name
        self.date_of_birth = date_of_birth
        self.email = email
        self.phone_number = phone_number

    def make_payment(self, amount, payment_date):
        self.payment_history.append({"amount": amount, "payment_date": payment_date})

    def get_enrolled_courses(self):
        return self.enrolled_courses

    def get_payment_history(self):
```

```
    return self.payment_history

def assign_teacher(self, teacher):
    self.teacher = teacher

def update_course_info(self, course_code, course_name, instructor):
    self.course_code = course_code
    self.course_name = course_name
    self.teacher = instructor

def get_enrollments(self):
    return self.enrollments

def get_teacher(self):
    return self.teacher



def get_student(self):
    return self.student

def get_course(self):
    return self.course
def update_teacher_info(self, first_name, last_name, email, expertise):
    self.first_name = first_name
    self.last_name = last_name
    self.email = email
    self.expertise = expertise

def get_assigned_courses(self):
    return self.assigned_courses
```

```
        self.payment_date = payment_date

    def get_student(self):
        return self.student

    def get_payment_amount(self):
        return self.amount

    def get_payment_date(self):
        return self.payment_date
```

Task 4

```
class DuplicateEnrollmentException(Exception):
    def __init__(self, message="Student is already enrolled in the course."):
        self.message = message
        super().__init__(self.message)

class CourseNotFoundException(Exception):
    def __init__(self, message="Course not found in the system."):
        self.message = message
        super().__init__(self.message)

class StudentNotFoundException(Exception):
    def __init__(self, message="Student not found in the system."):
        self.message = message
        super().__init__(self.message)

class TeacherNotFoundException(Exception):
    def __init__(self, message="Teacher not found in the system."):
        self.message = message
```

```
        super().__init__(self.message)

class PaymentValidationException(Exception):
    def __init__(self, message="Payment validation failed."):
        self.message = message
        super().__init__(self.message)

class InvalidStudentDataException(Exception):
    def __init__(self, message="Invalid student data."):
        self.message = message
        super().__init__(self.message)

class InvalidCourseDataException(Exception):
    def __init__(self, message="Invalid course data."):
        self.message = message
        super().__init__(self.message)

class InvalidEnrollmentDataException(Exception):
    def __init__(self, message="Invalid enrollment data."):
        self.message = message
        super().__init__(self.message)

class InvalidTeacherDataException(Exception):
    def __init__(self, message="Invalid teacher data."):
        self.message = message
        super().__init__(self.message)

class InsufficientFundsException(Exception):
    def __init__(self, message="Insufficient funds to enroll in the course."):
        self.message = message
        super().__init__(self.message)
```

Task 7

```
from datetime import datetime
import mysql.connector
from mysql.connector import Error

# Database connection
def create_connection():
    try:
        connection = mysql.connector.connect(
            host="localhost",
            user="root",
            password="root",
            port='3306',
            database="sisdb"
        )
        return connection
    except Error as e:
        print(f"Error connecting to the database: {e}")
        return None
```

Mains file

```
while True:
    print("\n=== Student Information System (SIS) Menu ===")
    print("1. Add New Student")
    print("2. Display Course Listings")
    print("3. Add Course Student")
    print("4. Add New Teacher")
    print("5. Update Teacher")
    print("6. Display Student Summary")
    print("7. Add New Payment")
    print("8. Display Enrollments")
```

```
    print("0. Exit")

    choice = input("Enter your choice: ")

    if choice == "1":
        add_new_student()
    elif choice == "2":
        display_course_listings()
    elif choice == "3":
        add_course_student()
    elif choice == "4":
        add_new_teacher()
    elif choice == "5":
        update_teacher()
    elif choice == "6":
        display_student_summary()
    elif choice == "7":
        add_new_payment()
    elif choice == "8":
        display_enroll()
    elif choice == "0":
        print("Exiting the program. Goodbye!")
        break
    else:
        print("Invalid choice. Please enter a valid option.")
```

Task 8

```
c = 20
```

```
def generate_stno():
    global c
    c+= 1
    return c


def add_new_student():
    connection = create_connection()
    if connection:
        try:
            studentid=generate_stno()
            firstname = input("Enter first name: ")
            lastname = input("Enter latname: ")
            date = input("Enter birthday date: ")
            email = input("Enter email: ")
            phone = input("Enter phone number: ")



            cursor = connection.cursor()
            cursor.execute("INSERT INTO students(studentid,firstname,lastname,date,email,phone)
VALUES (%s, %s, %s, %s,%s, %s)",
                        (studentid,firstname,lastname,date,email,phone))
            connection.commit()


            print("student details recorded successfully!")


        except (Error, ValueError) as e:
            print(f"Error recording student: {e}")
        finally:
            connection.close()
```

```
hallenge\petpals'; & 'C:\Users\Akilesh K\AppData\Local\Program
-2023.22.1\pythonFiles\lib\python\debugpy\adapter/../..\debugp
sign\sisdb\stduent.py'
Enter first name: John
Enter latname: Doe
Enter birthday date: 1995-08-15
Enter email: john.doe@example.com
Enter phone number: 123-456-7890
student details recorded successfully!
```

```
mysql> select * from students;
+-----------+-----------+-----------+------------+----------------------+----------------+
| studentid | firstname | lastname  | date       | email                | phone          |
+-----------+-----------+-----------+------------+----------------------+----------------+
|         1 | Gretchen  | Bird      | 1995-04-22 | bird@icloud.com      | (552) 253-2923 |
|         3 | Macy      | Travis    | 1990-01-29 | macy@outlook.com     | (157) 375-9678 |
|         5 | Jael      | Mcfarland | 1987-06-08 | jael@aol.couk        | (403) 558-6094 |
|         7 | Jamal     | Walls     | 1989-11-16 | jamal@yahoo.ca       | 1-344-653-4977 |
|         8 | Slade     | Boone     | 1999-03-28 | slade@aol.couk       | 1-377-644-1576 |
|        21 | John      | Doe       | 1995-08-15 | john.doe@example.com | 123-456-7890   |
+-----------+-----------+-----------+------------+----------------------+----------------+
6 rows in set (0.00 sec)
```

```
def display_course_listings():
    connection = create_connection()
    if connection:
        try:
            cursor = connection.cursor()
            cursor.execute("SELECT * FROM courses")
            cus = cursor.fetchall()

            print("courses:")
            for alist in cus:
                print(alist)

        except Error as e:
            print(f"Error retrieving  listings: {e}")
        finally:
            connection.close()
```

e = 1632

```
def generate_eno():
    global e
    e+= 1
    return e


def add_course_student():
    connection = create_connection()
    if connection:
        try:
            enrollementid=generate_eno()
            stude = input("Enter studentid: ")
            courseid = input("Enter course ID: ")
            date = datetime.now().strftime("%Y-%m-%d")



            cursor = connection.cursor()
            cursor.execute("INSERT INTO enrollments(enrollmentid,studentid,courseid,enrollmentdate)
VALUES (%s, %s, %s, %s)",
                        (enrollementid,stude,courseid,date))
            connection.commit()


            print("enrollment details recorded successfully!")


        except (Error, ValueError) as e:
            print(f"Error recording : {e}")
        finally:
            connection.close()
```

```
PS C:\Users\Akilesh K\OneDrive\Documents\python class\coding
hallenge\petpals'; & 'C:\Users\Akilesh K\AppData\Local\Progra
-2023.22.1\pythonFiles\lib\python\debugpy\adapter/../..\debug
sign\sisdb\stduent.py'
courses:
(32, 'english', 3, 367)
(57, 'maths', 4, 537)
(62, 'iot', 4, 642)
(82, 'french', 2, 846)
(88, 'chemistry', 3, 242)
Enter studentid: 21
Enter course ID: 57
enrollment details recorded successfully!
```

```
PS C:\Users\Akilesh K\OneDrive\Documents\python class\coding challe
hallenge\petpals'; & 'C:\Users\Akilesh K\AppData\Local\Programs\Pyt
-2023.22.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\lau
sign\sisdb\stduent.py'
courses:
(32, 'english', 3, 367)
(57, 'maths', 4, 537)
(62, 'iot', 4, 642)
(82, 'french', 2, 846)
(88, 'chemistry', 3, 242)
Enter studentid: 21
Enter course ID: 62
enrollment details recorded successfully!
```

```
mysql> select * from enrollments;
+--------------+-----------+----------+----------------+
| enrollmentid | studentid | courseid | enrollmentdate |
+--------------+-----------+----------+----------------+
|         1533 |        21 |       57 | 2023-12-25     |
|         1633 |        21 |       62 | 2023-12-25     |
|         3131 |         8 |       32 | 2023-09-21     |
|         4727 |         3 |       62 | 2023-08-08     |
|         5372 |         7 |       57 | 2023-11-03     |
|         7821 |         1 |       82 | 2023-10-12     |
|         8362 |         5 |       88 | 2023-09-16     |
+--------------+-----------+----------+----------------+
7 rows in set (0.01 sec)
```

Task 9

```
t = 20


def generate_tno():
    global t
    t+= 1
    return t


def add_new_teacher():
    connection = create_connection()
    if connection:
        try:
            teacherid=generate_tno()
            firstname = input("Enter first name: ")
            lastname = input("Enter latname: ")
            email = input("Enter email: ")





            cursor = connection.cursor()
            cursor.execute("INSERT INTO teachers(teacherid,firstname,lastname,email) VALUES (%s, %s, %s, %s)",
                           (teacherid,firstname,lastname,email))
            connection.commit()


            print("teacher details recorded successfully!")


        except (Error, ValueError) as e:
            print(f"Error recording teacher: {e}")
        finally:
            connection.close()
```

```
PS C:\Users\Akilesh K\OneDrive\Documents\python class
hallenge\petpals'; & 'C:\Users\Akilesh K\AppData\Loca
-2023.22.1\pythonFiles\lib\python\debugpy\adapter/../
sign\sisdb\stduent.py'
Enter first name: sarah
Enter latname: smith
Enter email: sarah.smith@example.com
teacher details recorded successfully!
```

```
mysql> select * from teachers;
+----------+----------+----------+-------------------------+
| teacherid | firstname | lastname | email                   |
+----------+----------+----------+-------------------------+
|        21 | sarah     | smith    | sarah.smith@example.com |
|       242 | Zahir     | Narayan  | new_email@example.com   |
|       367 | Owen      | Shan     | owen@google.org         |
|       537 | Allistair | Anand    | allistair9178@aol.couk  |
|       642 | Hermione  | Jindal   | hermione7644@aol.org    |
|       846 | Kevin     | Chandra  | kevin@aol.net           |
+----------+----------+----------+-------------------------+
6 rows in set (0.01 sec)
```

```
def update_teacher():
    connection = create_connection()
    if connection:
        try:
            cursor = connection.cursor()
            courseid=int(input("enter courseid:"))
            new_teacher=int(input("enter teacherid:"))
            update_query = ("UPDATE courses SET teacherid = %s WHERE courseid = %s")
            cursor.execute(update_query, (new_teacher, courseid))

            connection.commit()
```

```
        print(f"Updated value in the database.")

    except Error as e:
        print(f"Error updating value in the table: {e}")

    finally:
        connection.close()
```

```
PS C:\Users\Akilesh K\OneDrive\Documents\python
hallenge\petpals'; & 'C:\Users\Akilesh K\AppDat
-2023.22.1\pythonFiles\lib\python\debugpy\adapt
sign\sisdb\stduent.py'
enter courseid:32
enter teacherid:21
Updated value in the database.
```

```
mysql> select * from courses;
+----------+------------+---------+-----------+
| courseid | coursename | credits | teacherid |
+----------+------------+---------+-----------+
|       32 | english    |       3 |        21 |
|       57 | maths      |       4 |       537 |
|       62 | iot        |       4 |       642 |
|       82 | french     |       2 |       846 |
|       88 | chemistry  |       3 |       242 |
+----------+------------+---------+-----------+
5 rows in set (0.00 sec)
```

Task 10

```
def display_student_summary():
    connection = create_connection()
    if connection:
        try:
            cursor = connection.cursor()
```

```
        studentid=int(input("enter studentid ID:"))
        select_query=("SELECT * from students where studentid = %s ")
        cursor.execute(select_query,(studentid,))
        cdata = cursor.fetchone()

        print("student summary:")
        print(cdata)

    except Error as e:
        print(f"Error retrieving data: {e}")
    finally:
        connection.close()
```

```
PS C:\Users\Akilesh K\OneDrive\Documents\python class\coding challenge\petpals>  c:; cd 'c:
hallenge\petpals'; & 'C:\Users\Akilesh K\AppData\Local\Programs\Python\Python312\python.exe
-2023.22.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '49217' '--' 'c:\
sign\sisdb\stduent.py'
enter studentid ID:25
student summary:
(25, 'jane', 'johnson', datetime.date(1998, 6, 24), 'jane.johnson@gmail.com', '6876855')
```

```
def add_new_payment():
    connection = create_connection()
    if connection:
        try:
            paymentid=generate_pno()
            Studentid = input("Enter studentid: ")
            amount = input("Enter amount: ")
            date =datetime.now().strftime("%Y-%m-%d")
```

```
        cursor = connection.cursor()
        cursor.execute("INSERT INTO payments(paymentid,studentid,amount,paymentdate) VALUES (%s, %s, %s, %s)",
                (paymentid,Studentid,amount,date))
        connection.commit()


        print("teacher details recorded successfully!")


    except (Error, ValueError) as e:
        print(f"Error recording teacher: {e}")
    finally:
        connection.close()
```

```
mysql> select * from payments;
+-----------+-----------+--------+-------------+
| paymentid | studentid | amount | paymentdate |
+-----------+-----------+--------+-------------+
|        31 |        25 |    500 | 2023-12-25  |
|      3567 |         7 |   3000 | 2023-10-23  |
|      4342 |         8 |   5000 | 2023-10-23  |
|      5831 |         1 |   4000 | 2023-10-23  |
|      7362 |         3 |   5500 | 2023-10-23  |
|      9373 |         5 |   3500 | 2023-10-23  |
+-----------+-----------+--------+-------------+
6 rows in set (0.01 sec)
```

```
PS C:\Users\Akilesh K\OneDrive\Documents\pyth
hallenge\petpals'; & 'C:\Users\Akilesh K\AppD
-2023.22.1\pythonFiles\lib\python\debugpy\ada
sign\sisdb\stduent.py'
Enter studentid: 25
Enter amount: 500
teacher details recorded successfully!
```

TASK 11

```
def display_enroll():
    connection = create_connection()
    if connection:
        try:
            cursor = connection.cursor()
            courierid=int(input("enter course ID:"))
            select_query=("SELECT * from enrollments where courseid = %s ")
            cursor.execute(select_query,(courierid,))
            cdata = cursor.fetchall()


            print("enrollments :")
            for courierd in cdata:
                print(courierd)


        except Error as e:
            print(f"Error retrieving data: {e}")
        finally:
            connection.close()
```

```
PS C:\Users\Akilesh K\OneDrive\Documents\python cl
hallenge\petpals'; & 'C:\Users\Akilesh K\AppData\L
-2023.22.1\pythonFiles\lib\python\debugpy\adapter/
sign\sisdb\stduent.py'
enter course ID:62
enrollments :
(1633, 21, 62, datetime.date(2023, 12, 25))
(4727, 3, 62, datetime.date(2023, 8, 8))
```