```
# Database Connection
import pandas as pd
import mysql.connector
from sqlalchemy import create_engine
# Replace these values with your own database credentials

db_host = "localhost"
db_user = "root"
db_password = "toor"
db_name = "apex"

# Establish a connection to the database


print("The Software Is Created by : Team Apex")
print("")

menu_stack = []  # Stack to keep track of menu levels

while True:
   print("Choose The Number To Continue")
   print("1. Teachers")
   print("2. Students")
   print("3. exit")
   print("")
   fst_take = int(input("Enter The Option [1 or 2 or 3] [2 is default option] : ") or 2)

   if fst_take == 1:
      menu_stack.append(1)  # Push the menu level onto the stack
      while True:
         print("")
         print("1. View Teacher Detail")
         print("2. Add Teacher")
         print("3. Update Teacher")
         print("4. Remove Teacher")
         print("5. Salary Structure")
         print("6. Export Teachers Data to CSV")
         print("7. Back to Previous Menu")
         print("")

         tch_take_fst = int(input("Enter The Option : "))

         if tch_take_fst == 1:

            # View teacher details

            conn = mysql.connector.connect(
               host=db_host,
```

```
        user=db_user,
        password=db_password,
        database=db_name
    )

    cursor = conn.cursor()

    # Define the SQL query to fetch data from the table

    query = "SELECT * FROM tinfo"

    # Execute the query
    cursor.execute(query)

    # Fetch all the rows from the result set
    result = cursor.fetchall()

    # Display the fetched data
    for row in result:
        print("")
        print(row)
        print("")

    cursor.close()
    conn.close()

elif tch_take_fst == 2:
    # Add teacher

    conn = mysql.connector.connect(
        host=db_host,
        user=db_user,
        password=db_password,
        database=db_name
    )

    cursor = conn.cursor()

    # Define the INSERT INTO SQL statement
    insert_query = "INSERT INTO tinfo VALUES (%s, %s, %s, %s)"

    tname = input("Enter Teacher's Name : ")
    tfeild = input("Enter Teacher's Teaching Field : ")
    tnumb = input("Enter Teacher's Phone Number : ")
    sal = input("Enter Teacher's Salary : ")

    # Specify the values you want to insert
```

```
    values = (tname, tfeild, tnumb, sal)

    # Execute the query
    cursor.execute(insert_query, values)

    # Commit the changes to the database
    conn.commit()
    print("")
    print("done!")
    print("")

    cursor.close()
    conn.close()
elif tch_take_fst == 3:

    menu_stack_child_tch = []  # Stack to keep track of menu levels

    while True:

        print("Choose What You Want to Update.")
        print("")
        print("1. Teacher Name \n2. Teacher Salary \n3. Update Both \n4. Go To Previous Menu")
        print("")
        std_take_snd = int(input("Enter Option : "))

        if std_take_snd == 1:

            menu_stack_child_tch.append(1)  # Push the menu level onto the stack

            import mysql.connector

            # Connect to the database
            conn = mysql.connector.connect(
                host=db_host,
                user=db_user,
                password=db_password,
                database=db_name
            )

            cursor = conn.cursor()

            # Define the UPDATE SQL statement using a combination of columns to identify the row
            update_query = "UPDATE tinfo SET tname = %s WHERE tnumber = %s"

            new_name = input("Enter Your teacher's Name : ")
            teachers_number = int(input("Enter Your teachers Registerd Contact Number : "))
            # print("")
            # Specify the values you want to set in the UPDATE statement
```

```
    values = (new_name, teachers_number)

    # Execute the query to update the row
    cursor.execute(update_query, values)

    # Commit the changes to the database
    conn.commit()

    # Close the cursor and the connection
    cursor.close()
    conn.close()

    print("teachers information updated.")
    print("")
elif std_take_snd == 2:

    menu_stack_child_tch.append(2)  # Push the menu level onto the stack

    import mysql.connector

    # Connect to the database
    conn = mysql.connector.connect(
      host=db_host,
      user=db_user,
      password=db_password,
      database=db_name
    )

    cursor = conn.cursor()

    # Define the UPDATE SQL statement using a combination of columns to identify the row
    update_query = "UPDATE tinfo SET salary = %s WHERE tnumber = %s"

    new_class = input("Enter Your teacher's salary : ")
    teachers_number = int(input("Enter Your teacher's Registerd Contact Number : "))
    # print("")

    # Specify the values you want to set in the UPDATE statement
    values = (new_class, teachers_number)

    # Execute the query to update the row
    cursor.execute(update_query, values)

    # Commit the changes to the database
    conn.commit()

    # Close the cursor and the connection
    cursor.close()
```

```
    conn.close()

    print("teachers information updated.")

elif std_take_snd == 3:

    menu_stack_child_tch.append(3)  # Push the menu level onto the stack

    import mysql.connector

    # Connect to the database
    conn = mysql.connector.connect(
        host=db_host,
        user=db_user,
        password=db_password,
        database=db_name
    )

    cursor = conn.cursor()

    # Define the UPDATE SQL statement using a combination of columns to identify the row
    update_query = "UPDATE tinfo SET tname = %s, salary = %s WHERE tnumber = %s"

    new_name = input("Enter teacher New Name : ")
    new_class = input("Enter teacher New salary : ")

    teachers_number = int(
        input("Enter teacher's Registered Number : "))

    # Specify the values you want to set in the UPDATE statement
    values = (new_name, new_class, teachers_number)

    # Execute the query to update the row
    cursor.execute(update_query, values)

    # Commit the changes to the database
    conn.commit()

    # Close the cursor and the connection
    cursor.close()
    conn.close()

    print("teachers information updated.")
    print("")
elif std_take_snd == 4:

    if menu_stack_child_tch:
        # Go back to the previous menu if the list is not empty
```

```
                menu_stack_child_tch.pop()  # Remove the current menu level
            # else:
            #     print("Cannot go back, already at the top-level menu.")
            break
        else:
            print("INVALID ENTRY ! PLEASE CHECK YOUR OPTIONS")

    elif tch_take_fst == 4:

        # connection establishment From Database

        conn = mysql.connector.connect(
            host=db_host,
            user=db_user,
            password=db_password,
            database=db_name
        )

        cursor = conn.cursor()

        # This Vaariable is Use As identifier For Deleating Teachers

        numbb = int(input("Enter the Teacher Phone Number Which You Want To Remove From Database : "))

        # This is use to show the deleted table

        qry = f"SELECT * FROM tinfo WHERE tnumber = {numbb}"

        # Execute the query
        cursor.execute(qry)

        # Fetch all the rows from the result set
        result = cursor.fetchall()

        # Display the fetched data
        for row in result:
            print("")
            print(row)
            print("")
            print("Deleted!")

        query = f"DELETE FROM tinfo WHERE tnumber = {numbb}"

        # Execute the query to delete rows
        cursor.execute(query)

        # Commit the changes to the database
```

```
            conn.commit()

            # Close the cursor and the connection
            cursor.close()
            conn.close()

        elif tch_take_fst == 5:

            # this value is imported from sal.py file
            # print(sal.salary)

            import globe_var

            data = globe_var.salary
            df = pd.Series(data)
            print(df)

        elif tch_take_fst == 6:

            # Establish a connection to the database
            conn = mysql.connector.connect(
                host=db_host,
                user=db_user,
                password=db_password,
                database=db_name
            )

            # Create an SQLAlchemy engine using the MySQL connection
            engine =
create_engine(f"mysql+mysqlconnector://{db_user}:{db_password}@{db_host}/{db_name}")

            # Replace this query with your own SQL query
            sql_query = "SELECT * FROM tinfo"

            # Fetch the data into a DataFrame
            df = pd.read_sql_query(sql_query, engine)

            # Close the database connection
            conn.close()

            # Specify the CSV file path where you want to save the data
            csv_file_path = "tdata.csv"

            # Save the data to a CSV file
            df.to_csv(csv_file_path, )  # we can use if we don't need indexes : index=false
            print("")
            print("Exported!")
```

```
        elif tch_take_fst == 7:
            # Go back to the previous menu
            menu_stack.pop()  # Remove the current menu level
            break

        else:
            print("INVALID ENTRY ! PLEASE CHECK YOUR OPTIONS")

elif fst_take == 2:

    menu_stack.append(2)  # Push the menu level onto the stack
    while True:
        print("")
        print("1. View Students Detail")
        print("2. Add Student")
        print("3. Update Student")
        print("4. Remove Student")
        print("5. Fees Structure")
        print("6. Export Students Data to CSV")
        print("7. Back to Previous Menu")
        print("")

        std_take_fst = int(input("Enter The Option : "))

        if std_take_fst == 1:

            # View teacher details

            conn = mysql.connector.connect(
                host=db_host,
                user=db_user,
                password=db_password,
                database=db_name
            )

            cursor = conn.cursor()

            # Define the SQL query to fetch data from the table

            query = "SELECT * FROM sinfo"

            # Execute the query
            cursor.execute(query)

            # Fetch all the rows from the result set
            result = cursor.fetchall()

            # Display the fetched data
```

```
            for row in result:
                print("")
                print(row)
                print("")

            cursor.close()
            conn.close()

        elif std_take_fst == 2:
            # Add student

            # Connect to the database
            conn = mysql.connector.connect(
                host=db_host,
                user=db_user,
                password=db_password,
                database=db_name
            )

            cursor = conn.cursor()

            # Define the INSERT INTO SQL statement, specifying column names except for 'id'
            insert_query = "INSERT INTO sinfo (sname, sdob, snumber, sclass, address) VALUES (%s, %s,
%s, %s, %s)"

            sname = input("Enter Student's Name: ")
            sdob = input("Enter Student dob: ")
            snumber = input("Enter Student Phone Number: ")
            sclass = input("Enter Student Class: ")
            address = input("Enter Student Address: ")

            # Specify the values you want to insert
            values = (sname, sdob, snumber, sclass, address)

            # Execute the query
            cursor.execute(insert_query, values)

            # Commit the changes to the database
            conn.commit()
            print("")
            print("Done!")
            print("")

            # Close the cursor and connection
            cursor.close()
            conn.close()

        elif std_take_fst == 3:
```

```
menu_stack_child = []  # Stack to keep track of menu levels

while True:

    print("Choose What You Want to Update.")
    print("")
    print("1. Student Name \n2. Student Class \n3. Update Both \n4. Go To Previous Menu")
    print("")
    std_take_snd = int(input("Enter Option : "))

    if std_take_snd == 1:

        menu_stack_child.append(1)  # Push the menu level onto the stack

        import mysql.connector

        # Connect to the database
        conn = mysql.connector.connect(
            host=db_host,
            user=db_user,
            password=db_password,
            database=db_name
        )

        cursor = conn.cursor()

        # Define the UPDATE SQL statement using a combination of columns to identify the row
        update_query = "UPDATE sinfo SET sname = %s WHERE snumber = %s"

        new_name = input("Enter Your Student Name : ")
        student_number = int(input("Enter Your Student Registerd Contact Number : "))
        # print("")
        # Specify the values you want to set in the UPDATE statement
        values = (new_name, student_number)

        # Execute the query to update the row
        cursor.execute(update_query, values)

        # Commit the changes to the database
        conn.commit()

        # Close the cursor and the connection
        cursor.close()
        conn.close()

        print("Student information updated.")
        print("")
```

```
elif std_take_snd == 2:

    menu_stack_child.append(2)  # Push the menu level onto the stack

    import mysql.connector

    # Connect to the database
    conn = mysql.connector.connect(
        host=db_host,
        user=db_user,
        password=db_password,
        database=db_name
    )

    cursor = conn.cursor()

    # Define the UPDATE SQL statement using a combination of columns to identify the row
    update_query = "UPDATE sinfo SET sclass = %s WHERE snumber = %s"

    new_class = input("Enter Your Student class : ")
    student_number = int(input("Enter Your Student Registerd Contact Number : "))
    # print("")

    # Specify the values you want to set in the UPDATE statement
    values = (new_class, student_number)

    # Execute the query to update the row
    cursor.execute(update_query, values)

    # Commit the changes to the database
    conn.commit()

    # Close the cursor and the connection
    cursor.close()
    conn.close()

    print("Student information updated.")

elif std_take_snd == 3:

    menu_stack_child.append(3)  # Push the menu level onto the stack

    import mysql.connector

    # Connect to the database
    conn = mysql.connector.connect(
        host=db_host,
        user=db_user,
```

```
                    password=db_password,
                    database=db_name
                )

                cursor = conn.cursor()

                # Define the UPDATE SQL statement using a combination of columns to identify the row
                update_query = "UPDATE sinfo SET sname = %s, sclass = %s WHERE snumber = %s"

                new_name = input("Enter Student New Name : ")
                new_class = input("Enter Student New Class : ")

                student_number = int(
                    input("Enter Your Student Registered Number : "))  # Replace with the actual phone
number

                # Specify the values you want to set in the UPDATE statement
                values = (new_name, new_class, student_number)

                # Execute the query to update the row
                cursor.execute(update_query, values)

                # Commit the changes to the database
                conn.commit()

                # Close the cursor and the connection
                cursor.close()
                conn.close()

                print("Student information updated.")
                print("")
            elif std_take_snd == 4:

                if menu_stack_child:
                    # Go back to the previous menu if the list is not empty
                    menu_stack_child.pop()  # Remove the current menu level
                # else:
                #     print("Cannot go back, already at the top-level menu.")
                break
            else:
                print("INVALID ENTRY ! PLEASE CHECK YOUR OPTIONS")

        elif std_take_fst == 4:

            # Remove student

            # connection establishment From Database
```

```
            conn = mysql.connector.connect(
                host=db_host,
                user=db_user,
                password=db_password,
                database=db_name
            )

            cursor = conn.cursor()

            # This Variable is Use As identifier For Deleating Teachers

            numbb = int(input("Enter the Student Phone Number Which You Want To Remove From
Database : "))

            # This is use to show the deleted table

            qry = f"SELECT * FROM sinfo WHERE snumber = {numbb}"

            # Execute the query
            cursor.execute(qry)

            # Fetch all the rows from the result set
            result = cursor.fetchall()

            # Display the fetched data
            for row in result:
                print("")
                print(row)
                print("")
                print("Deleted!")
                print("")

            query = f"DELETE FROM sinfo WHERE snumber = {numbb}"

            # Execute the query to delete rows
            cursor.execute(query)

            # Commit the changes to the database
            conn.commit()

            # Close the cursor and the connection
            cursor.close()
            conn.close()

        elif std_take_fst == 5:

            import globe_var
```

```
            data = globe_var.fee
            df = pd.Series(data)
            print(df)


        elif std_take_fst == 6:

            # Establish a connection to the database
            conn = mysql.connector.connect(
                host=db_host,
                user=db_user,
                password=db_password,
                database=db_name
            )

            # Create an SQLAlchemy engine using the MySQL connection
            engine =
create_engine(f"mysql+mysqlconnector://{db_user}:{db_password}@{db_host}/{db_name}")

            # Replace this query with your own SQL query
            sql_query = "SELECT * FROM sinfo"

            # Fetch the data into a DataFrame
            df = pd.read_sql_query(sql_query, engine)

            # Close the database connection
            conn.close()

            # Specify the CSV file path where you want to save the data
            csv_file_path = "sdata.csv"

            # Save the data to a CSV file
            df.to_csv(csv_file_path, )  # we can use if we don't need indexes : index=false
            print("")
            print("Exported!")

        elif std_take_fst == 7:

            # Go back to the previous menu
            menu_stack.pop()  # Remove the current menu level

            break
        else:
            print("INVALID ENTRY ! PLEASE CHECK YOUR OPTIONS")

    elif fst_take == 3:
        import sys
```

```
    print("Exiting the program. Goodbye!")
    sys.exit()  # Use sys.exit() to exit the program
else:
    print("INVALID ENTRY!")
```