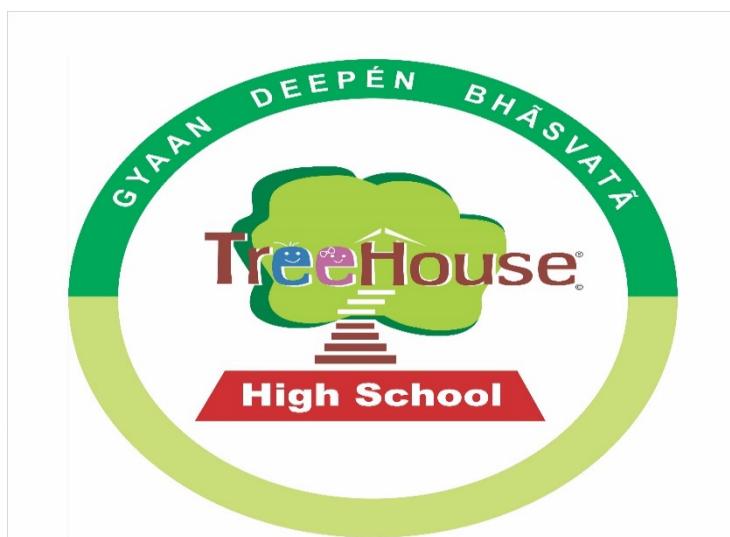


Tree House High School (CBSE)
Near Vuda Colony, Opp. Pramukh Bazaar, Atladara, Vadodara(GJ)

A Project Report on
Employee Management System
Subject: Computer Science
Academic Year: 2024-25



Tree House High School (CBSE)

Submitted By: Ojas Shailesh Deshpande

Class: XII – B

Roll no:

Under Guidance of:

Ms. Nikisha Patel
PGT(Computer Science)

CERTIFICATE

This is to certify that **Ojas Shailesh Deshpande** of class: **XII B** has prepared the report on the project entitled "**Employee Management System**" for academic year 2024-25 . The report is the result of his/her efforts & endeavours. The report is found worthy acceptance as final project report for the subject Computer Science of Class XII. He/She has prepared the report under my guidance.

Internal Examiner

Ms. Nikisha Patel
PGT(Computer Science)

External Examiner

Principal

Dr. Bharat Bhushan Vyas

ACKNOWLEDGMENT

I would like to express a deep sense of thanks & gratitude to my project guide Ms. Nikisha Patel Mam for guiding me immensely through the course of the project. She always evinced keen interest in my work. Her constructive advice & constant motivation have been responsible for the successful completion of this project.

My sincere thanks goes to Dr. Bharat Bhushan Vyas , Our Principal sir, for his co-ordination in extending every possible support for the completion of this project.

I also thanks to my parents for their motivation & support. I must thanks to my classmates for their timely help & support for compilation of the project.

Last but not the least , I would like to thank all those who had helped directly or indirectly towards the completion of the project.

Student Name: Ojas Shailesh Deshpande

Class: XII – B

INDEX

Content	Page No
Certificate	2
Acknowledgement	3
Introduction	5
Hardware & software Requirements	6
MySQL Code	7
Python Code	8
Output	12
Limitations	15
Enhancements	16
Bibliography	17

INTRODUCTION

The project is designed to keep records of Employee's who are working in the companies. A table named Employee is created in the database Employee_Data using MySQL to store information about Employee Name, Contact, Address, Post, Date of Birth, Salary with Employee ID and Joining Date. Administrator of the project can enter new record, Display all or specific employee record also can modify and delete records in the table.

HARDWARE & SOFTWARE REQUIREMENTS

1. RAM: 16 GB
2. Hard disk : 100 GB HDD
3. OPERATING SYSTEM: WINDOWS 7 AND ABOVE
4. Python 3.13.0
5. MySQL Server must be installed on the system.
6. MySQL-Python connector must be installed.

MySQL CODE

```
mysql> create database Employee_Data;  
Query OK, 1 row affected (0.02 sec)
```

```
mysql> use Employee_Data;  
Database changed
```

```
mysql> Create table Employee(  
-> EID INT NOT NULL,  
-> EJoining_Date DATE NOT NULL,  
-> EName VARCHAR(50) NOT NULL,  
-> EPhone_Number BIGINT NOT NULL,  
-> EDOB DATE NOT NULL,  
-> EAddress VARCHAR(200) NOT NULL,  
-> EPost VARCHAR(20) NOT NULL,  
-> ESalary INT NOT NULL,  
-> PRIMARY KEY (EID));
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> describe Employee;
```

Field	Type	Null	Key	Default	Extra
EID	int	NO	PRI	NULL	
EJoining_Date	date	NO		NULL	
EName	varchar(50)	NO		NULL	
EPhone_Number	bigint	NO		NULL	
EDOB	date	NO		NULL	
EAddress	varchar(200)	NO		NULL	
EPost	varchar(20)	NO		NULL	
ESalary	int	NO		NULL	

```
8 rows in set (0.01 sec)
```

Python CODE

```
# Employee Management System, By Ojas Deshpande

from os import system
import mysql.connector
con = mysql.connector.connect(host="localhost", user="root", password="12345678", database="Employee_Data")

# Function to Add Employee
def Add_Employee():
    print("\nAdd Employee's Record-->")
    EID = input ("\nEnter Employee's ID:")
    # Checking if Employee Id is existing
    if(check_employee(EID) == False):
        EJoining_Date = int(input("Enter Employee's Date of Joining [YYYYMMDD]:"))
        if len(str(EJoining_Date)) == 8:
            EName = input("Enter Employee's Name:")
            if len(EName) <= 50:
                EPhone_Number = input("Enter Employee's Phone Number:")
                if len(EPhone_Number) == 10:
                    if (check_employee_EPhone_Number(EPhone_Number) == False):
                        EDOB = int(input("Enter Employee's Date of Birth [YYYYMMDD]:"))
                        if len(str(EDOB)) == 8:
                            EAddress = input("Enter Employee's Address:")
                            if len(EAddress) <= 200:
                                EPost = input("Enter Employee's Post:")
                                if len(EPost) <= 20:
                                    ESalary = input("Enter Employee's Salary:")
                                    data = (EID, EJoining_Date, EName, EPhone_Number, EDOB, EAddress, EPost, ESalary)
                                    # Inserting Employee Details into the Employee_Data (Employee) Table
                                    sql = 'insert into Employee values(%s,%s,%s,%s,%s,%s,%s,%s)'
                                    c = con.cursor()
                                    c.execute(sql, data)
                                    con.commit()
                                    print("\nSuccessfully Added Employee's Record.")
                                    press = input("\nPress Any Key To Continue.")
                                    menu()
                                else:
                                    print("\nPost is too Long!")
                                    press = input ("\\nPress Any Key To Continue.")
                                    menu()
                            else:
                                print("\nAddress is too Long!")
                                press = input ("\\nPress Any Key To Continue.")
                                menu()
                        else:
                            print("\nDate of Birth is Invalid!")
                            press = input ("\\nPress Any Key To Continue.")
                            menu()
                    else:
                        print("\nPhone Number Already Exists!")
                        press = input ("\\nPress Any Key To Continue.")
                        menu()
                else:
                    print("\nPhone Number is Invalid!")
                    press = input ("\\nPress Any Key To Continue.")
                    menu()
            else:
                print("\nName is too long!")
                press = input ("\\nPress Any Key To Continue.")
                menu()
        else:
            print("\nDate of Joining is invalid!")
            press = input ("\\nPress Any Key To Continue.")
            menu()
    else:
        print("\nEmployee ID Already Exists!")
        press = input ("\\nPress Any Key To Continue.")
        menu()
```

```

# Function to Check if Employee with given EPhone_Number exists
def check_employee_EPhone_Number(employee_phone_no):
    sql = 'select * from Employee where EPhone_Number = %s'
    c = con.cursor(buffered=True)
    data = (employee_phone_no,)
    c.execute(sql, data)
    a = c.rowcount
    if a == 1:
        return True
    else:
        return False

# Function to Check if Employee with given EID exists
def check_employee(employee_id):
    sql = 'select * from Employee where EID = %s'
    c = con.cursor(buffered=True)
    data = (employee_id,)
    c.execute(sql, data)
    a = c.rowcount
    if a == 1:
        return True
    else:
        return False

# Function to Display Employee
def Display_Employee():
    print("\nDisplay Employee's Record--> ")
    # Query to select all rows from Employee_Data (Employee) Table
    sql = 'select * from Employee'
    c = con.cursor()
    c.execute(sql)
    # Fetching all details of all the Employees
    r = c.fetchall()
    a = c.rowcount
    if a == 0:
        print("\nEmployee Record does not Exist!")
        press = input("\nPress Any Key To Continue.")
        menu()
    else:
        for i in r:
            print("\nEmployee's Id: ", i[0])
            print("Employee's Joining Date: ", i[1])
            print("Employee's Name: ", i[2])
            print("Employee's Phone Number: ", i[3])
            print("Employee's Date of Birth: ", i[4])
            print("Employee's Address: ", i[5])
            print("Employee's Post: ", i[6])
            print("Employee's Salary: ", i[7])
    else:
        press = input("\nPress Any Key To Continue.")
        menu()

# Function to Update Employee
def Update_Employee():
    print("\nUpdate Employee's Record-->")
    EID = input("\nEnter Employee's ID: ")
    # Checking If Employee Id exists
    if (check_employee(EID) == True):
        EPhone_Number = input("Enter Employee's Phone Number: ")
        if len(EPhone_Number) == 10:
            if (check_employee_EPhone_Number(EPhone_Number) == False):
                EAddress = input("Enter Employee's Address: ")
                if len(EAddress) <= 200:
                    # Updating Employee details in Employee_Data (Employee) Table
                    sql = 'UPDATE Employee SET EPhone_Number = %s, EAddress = %s WHERE EID = %s'
                    data = (EPhone_Number, EAddress, EID)
                    c = con.cursor()
                    c.execute(sql, data)
                    con.commit()
                    print("\nUpdated Employee's Record")
                    press = input("\nPress Any Key To Continue.")
                    menu()

```

```

        else:
            print("\nAddress is too Long!")
            press = input ("\nPress Any Key To Continue.")
            menu()

        else:
            print("\nPhone Number Already Exists!")
            press = input ("\nPress Any Key To Continue.")
            menu()

        else:
            print("\nPhone Number is Invalid!")
            press = input ("\nPress Any Key To Continue.")
            menu()

        else :
            print("\nEmployee Record does not exist!")
            press = input("\nPress Any Key To Continue.")
            menu()

# Function to Remove_Employee
def Remove_Employee():
    print("\nRemove Employee's Record-->")
    EID = input("\nEnter Employee's ID:")
    # Checking If Employee Id is exists
    if (check_employee(EID) == False):
        print("\nEmployee's Record does not exist!")
        press = input("\nPress Any Key To Continue.")
        menu()
    else:
        # Query to delete Employee from Employee table
        sql = 'delete from Employee where EID = %s'
        data = (EID,)
        c = con.cursor()
        c.execute(sql, data)
        con.commit()
        print("\nEmployee Removed!")
        press = input("\nPress Any key To Continue.")
        menu()

# Function to Search Employee
def Search_Employee():
    print("\nSearch Employee's Record-->")
    EID = input("\nEnter Employee ID: ")
    # Checking If Employee Id exists
    if (check_employee(EID) == True):
        # Query to search Employee from Employee table
        sql = 'select * from Employee where EID = %s'
        data = (EID,)
        c = con.cursor()
        c.execute(sql, data)
        # Fetching all details of all the employee
        r = c.fetchall()
        for i in r:
            print("\nEmployee's Id: ", i[0])
            print("Employee's Joining Date: ", i[1])
            print("Employee's Name: ", i[2])
            print("Employee's Phone Number: ", i[3])
            print("Employee's Date of Birth: ", i[4])
            print("Employee's Address: ", i[5])
            print("Employee's Post: ", i[6])
            print("Employee's Salary: ", i[7])
        press = input("\nPress Any key To Continue.")
        menu()
    else:
        print("\nEmployee Record does not Exist!")
        press = input("\nPress Any Key To Continue.")
        menu()

```

```

# Menu function to display menu
def menu():
    print("-----")
    print("Employee Management System")
    print("-----")
    print(" 1. Add Employee")
    print(" 2. Display Employee's Record")
    print(" 3. Update Employee's Record")
    print(" 4. Remove Employee's Record")
    print(" 5. Search Employee's Record")
    print(" 6. Exit")
    print("-----")
    print("Choice Options: [1/2/3/4/5/6]:")
    print("-----")

ch = int(input("Enter your Choice: "))
if ch == 1:
    Add_Employee()
elif ch == 2:
    Display_Employee()
elif ch == 3:
    Update_Employee()
elif ch == 4:
    Remove_Employee()
elif ch == 5:
    Search_Employee()
elif ch == 6:
    print("THANK YOU!")
    exit(0)
else:
    print("\nInvalid Choice!")
    press = input("\nPress Any key To Continue.")
    menu()

menu()

```

OUTPUT

```
Python 3.13.0 (v3.13.0:60403a5409f, Oct 7 2024, 00:37:40) [Clang 15.0.0 (clang-1500.3.9.4)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.

>>> = RESTART: /Users/ojasdeshpande/Desktop/Education/Projects Class 12/Computer Science/Employee management system
.py
-----
Employee Management System
-----
1. Add Employee
2. Display Employee's Record
3. Update Employee's Record
4. Remove Employee's Record
5. Search Employee's Record
6. Exit
-----
Choice Options: [1/2/3/4/5/6]:
-----
Enter your Choice:
```



```
Enter your Choice: 1
Add Employee's Record-->
Enter Employee's ID:01
Enter Employee's Date of Joining [YYYYMMDD]:20241007
Enter Employee's Name:Ojas Deshpande
Enter Employee's Phone Number:1234567890
Enter Employee's Date of Birth [YYYYMMDD]:20201209
Enter Employee's Address:Vadodara
Enter Employee's Post:Head
Enter Employee's Salary:100000
Successfully Added Employee's Record.
Press Any Key To Continue.
```



```
Enter your Choice: 2
Display Employee's Record-->
Employee's Id: 1
Employee's Joining Date: 2024-10-07
Employee's Name: Ojas Deshpande
Employee's Phone Number: 1234567890
Employee's Date of Birth: 2020-12-09
Employee's Address: Vadodara
Employee's Post: Head
Employee's Salary: 100000
Press Any Key To Continue.
```

Enter your Choice: 3

Update Employee's Record-->

Enter Employee's ID: 01

Enter Employee's Phone Number: 9021345678

Enter Employee's Address: Gujarat

Updated Employee's Record

Press Any Key To Continue.

Employee Management System

1. Add Employee
2. Display Employee's Record
3. Update Employee's Record
4. Remove Employee's Record
5. Search Employee's Record
6. Exit

Choice Options: [1/2/3/4/5/6]:

Enter your Choice: 2

Display Employee's Record-->

Employee's Id: 1

Employee's Joining Date: 2024-10-07

Employee's Name: Ojas Deshpande

Employee's Phone Number: 9021345678

Employee's Date of Birth: 2020-12-09

Employee's Address: Gujarat

Employee's Post: Head

Employee's Salary: 100000

Press Any Key To Continue.

Enter your Choice: 5

Search Employee's Record-->

Enter Employee ID: 01

Employee's Id: 1

Employee's Joining Date: 2024-10-07

Employee's Name: Ojas Deshpande

Employee's Phone Number: 9021345678

Employee's Date of Birth: 2020-12-09

Employee's Address: Gujarat

Employee's Post: Head

Employee's Salary: 100000

Press Any key To Continue.

Enter your Choice: 4

Remove Employee's Record-->

Enter Employee's ID:01

Employee Removed!

Press Any key To Continue.

Employee Management System

- 1. Add Employee
 - 2. Display Employee's Record
 - 3. Update Employee's Record
 - 4. Remove Employee's Record
 - 5. Search Employee's Record
 - 6. Exit
-

Choice Options: [1/2/3/4/5/6]:

Enter your Choice: 2

Display Employee's Record-->

Employee Record does not Exist!

Press Any Key To Continue.

Employee Management System

- 1. Add Employee
 - 2. Display Employee's Record
 - 3. Update Employee's Record
 - 4. Remove Employee's Record
 - 5. Search Employee's Record
 - 6. Exit
-

Choice Options: [1/2/3/4/5/6]:

Enter your Choice: 6

THANK YOU!

LIMITATIONS

1. **Scalability:** Python and MySQL may not be the most scalable options for very large organizations.
2. **Security:** Python and MySQL are both open-source, which means that they are more vulnerable to security risks.

ENHANCEMENTS

1. Add more functions to the system.
2. **Improve security:** Implement robust security measures to protect sensitive employee data.
3. **Data Integrity:**
 - Implement data validation and constraints to ensure that only valid data is entered into the database.
 - Use triggers to enforce business rules and maintain data consistency .
4. **Data Backup and Recovery:** Regularly back up the database to prevent data loss.

BIBLIOGRAPHY

1. Google search engine.
2. Computer Science with Python, Class 12 By Sumita Arora.