KONGU ENGINEERING COLLEGE
(Autonomous)
Perundurai, Erode – 638 060

DEPARTMENT OF COMPUTER APPLICATIONS

AN APPLICATION PROJECT REPORT For EMPLOYEE MANAGEMENT SYSTEM

PYTHON PROGRAMMING (22MCC11)

Submitted by

AMANRAJ
(22MCR004)

KONGU ENGINEERING COLLEGE

(Autonomous)

Perundurai, Erode – 638 060

DEPARTMENT OF COMPUTER APPLICATIONS

BONAFIDE CERTIFICATE

NAME : **AMAN RAJ (22MCR004)**
MAHASHWETHA.G S (22MCR060)
MARIA JOHN PAUL.M(22MCR062)

Course Code : **22MCC11**

Course Name : **PYTHON PROGRAMMING**

Semester : **I**

Certified that this is a bonafide record of work for application project done by the above students for **22MCC11 – PYTHON PROGRAMMING** during the academic year **2022-2023.**

Submitted for the Viva Voce Examination held on _____

Lab-in-Charge **Head of the Department**

INDEX

S.NO	TITLE	PAGE No.
1	ABSTRACT	1
2	MODULE DESCRIPITION	2
3	DATABASE DESIGN	3
4	SOURCE CODE	4
5	SCREENSHOT	16
6	CONCLUSION	20

ABSTRACT

Employee Management system is an application that enables users to create and store Employee Records. This system protects the professional and personal information of employees and the Company. This System works on the concepts of taking the information from the database making required changes in the fetched data and applying the changes in the record which we will see in our Promote Employee System. We can also have the information about all the existing employees by using the Display Employee function. It assists in the automation of manual operations, saving both time and money. The main advantage of connecting our program to the database is that the information becomes lossless even after closing our program a number of times.

It also helps to know the number of employees in different departments .This application is helpful to department of the organization which maintains data of employees related to an organization.It is simple to understand and can be used by anyone who is not even familiar with simple employees system. It is user friendly and just asks the user to follow step by step operations by giving him few options. It is fast and can perform many operations of a company.It helps to have full –fledged control over his/her employees.It receives user name and password to login and register.It keeps the record of user salary,department and employee name , age and so on....

This software package has been developed using the powerful coding tools of PYTHON. The package contains different modules like Employee details. This version of the software has multi-user approach. For further enhancement or development of the package, user feedback will be considered.

APPLICATION MODULES

- Add Employee
- Update Employee
- Delete Employee

ADD EMPLOYEE:

This module allows the organization to add the new employee details to the employee database. The employee details will be displayed in the bottom of the page.

UPDATE EMPLOYEE:

This module allows the organization to update the details of the employee. These details are updated in the employee database.

DELETE EMPLOYEE:

In this module, organization can delete the particular employee details in the employee database. Details of the particular employee are removed in the database.

CODING

#Update

```
    def update_data(self):
        if self.var_dep.get()==""or self.var_email.get()=="":
            messagebox.showerror('Error','All Fields are required')
        else:
            try:
                update=messagebox.askyesno('update','Are you sure update this
employee data')
                if update>0:

conn=mysql.connector.connect(host='localhost',username='root',password='
Sindi',database='employee')
                    my_cursor=conn.cursor()
                    my_cursor.execute('update employee set
Department=%s,Name=%s,Designition=%s,Email=%s,Address=%s,Married
_status=%s,DOB=%s,DOJ=%s,ID_TYPE=%s,Gender=%s,Phone_no=%s,C
ountry=%s,Salary=%s where ID_NUM=%s',(

                            self.var_dep.get(),
                            self.var_name.get(),
                            self.var_designation.get(),
                            self.var_email.get(),
                            self.var_address.get(),
                            self.var_married.get(),
                            self.var_dob.get(),
                            self.var_doj.get(),
                            self.var_idproffcomb.get(),
                            self.var_gender.get(),
                            self.var_phone.get(),
                            self.var_country.get(),
                            self.var_salary.get(),
                            self.var_idproff.get(),
                                        ))
                else:
                    if not update:
                        return
                conn.commit()
                self.fetch_data()
                conn.close()
```

```
            messagebox.showinfo('suceess','Employee Successfully
update',parent=self.root)
        except Exception as es:
            messagebox.showerror('Error',f'Due To:{str(es)}',parent=self.root)
```

#Delete

```
    def delete_data(self):
        if self.var_dep.get()==""or self.var_email.get()=="":
            messagebox.showerror('Error','All Fields are required')
        else:
            try:
                Delete=messagebox.askyesno('Delete','Are you sure delete this
employee',parent=self.root)
                if Delete>0:

conn=mysql.connector.connect(host='localhost',username='root',password='
Sindi',database='employee')
                    my_cursor=conn.cursor()
                    sql='delete from employee where ID_NUM=%s'
                    value=(self.var_idproff.get(),)
                    my_cursor.execute(sql,value)
                else:
                    if not Delete:
                        return
                conn.commit()
                self.fetch_data()
                conn.close()
                messagebox.showinfo('Delete','Employee Successfully
Deleted',parent=self.root)
            except Exception as es:
                messagebox.showerror('Error',f'Due To:{str(es)}',parent=self.root)
```

#Reset

```
    def reset_data(self):

        self.var_dep.set("select Department")
        self.var_name.set("")
        self.var_designation.set("")
```

```
        self.var_email.set("")
        self.var_address.set("")
        self.var_married.set("Married")
        self.var_dob.set("")
        self.var_doj.set("")
        self.var_idproffcomb.set("select ID")
        self.var_idproff.set("")
        self.var_gender.set("")
        self.var_phone.set("")
        self.var_country.set("")
        self.var_salary.set("")
```

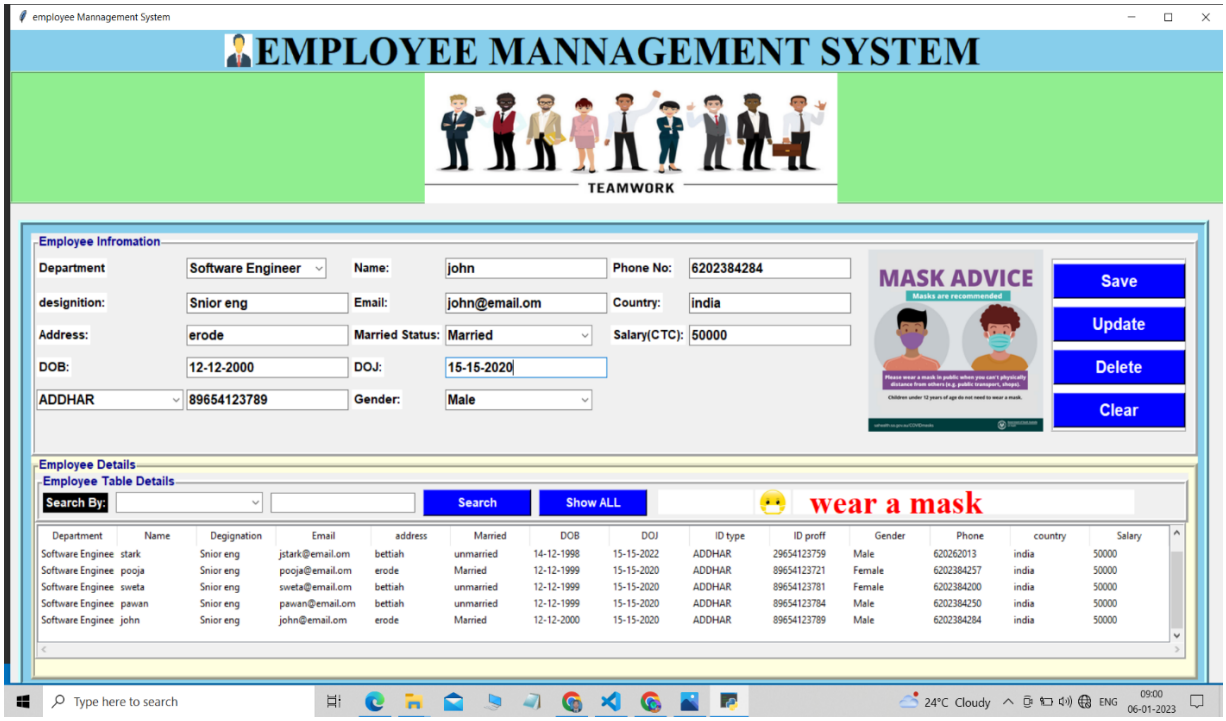
#Search

```
    def search_data(self):
        if self.var_com_search.get()=="or self.var_search.get()=="":
            messagebox.showerror('Error','Please select option')
        else:
            try:

conn=mysql.connector.connect(host='localhost',username='root',password='
Sindi',database='employee')
                my_cursor=conn.cursor()
                my_cursor.execute('select *from employee where '
+str(self.var_com_search.get())+" LIKE
'%"+str(self.var_search.get()+"%'"))
                rows=my_cursor.fetchall()
                if len(rows)!=0:
                    self.employee_table.delete(*self.employee_table.get_children())
                    for i in rows:
                        self.employee_table.insert("",END,values=i)
                conn.commit()
                conn.close()
            except Exception as es:
                 messagebox.showerror('Error',f'Due
To:{str(es)}',parent=self.root)




if __name__=="__main__":
    root=Tk()
```

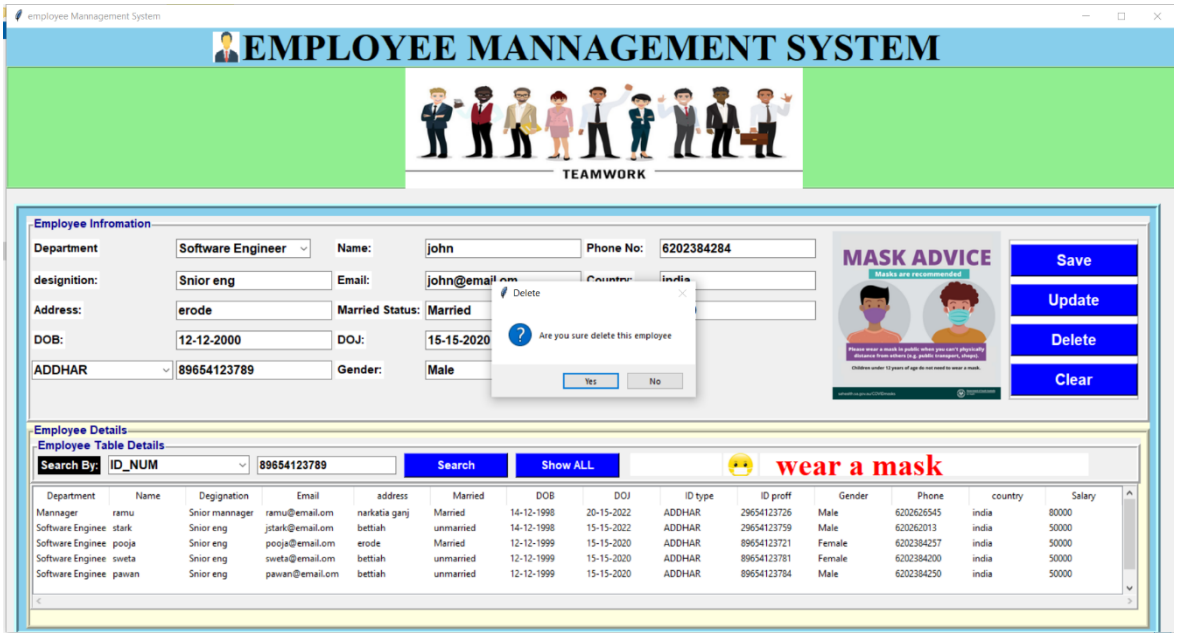
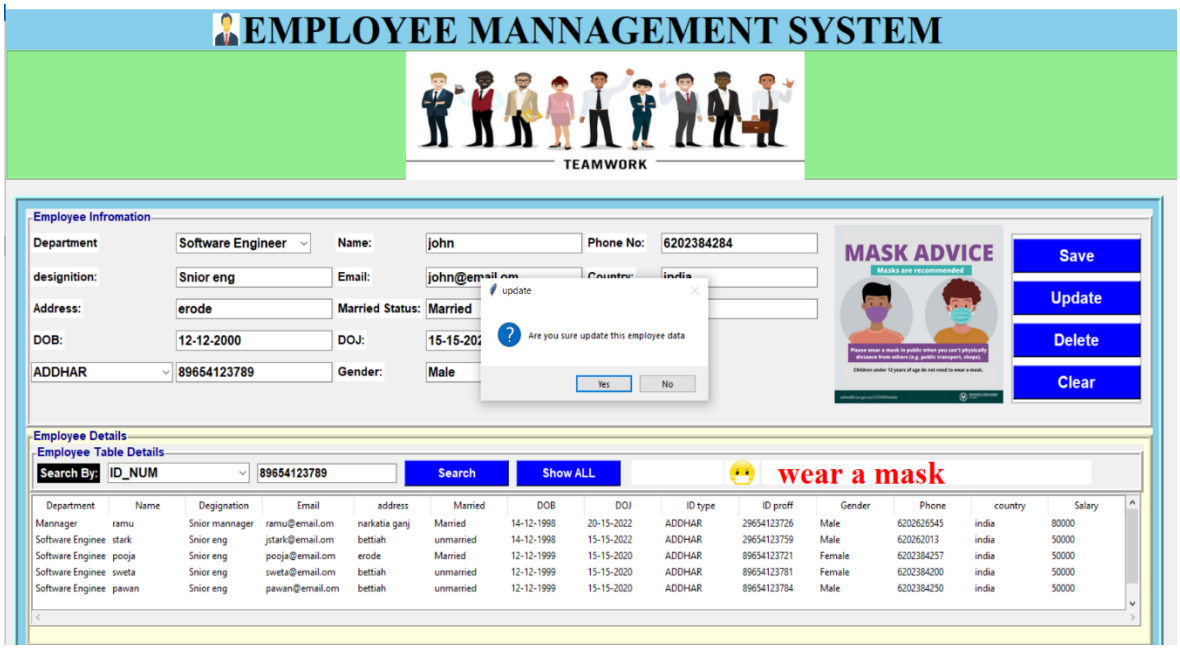
```
obj=Employee(root)
root.mainloop()
```

Search Details

Show all Employee details



Deleting Details



Update Details