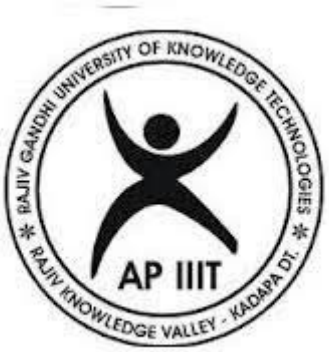
Project Report
on
EMPLOYEE MANAGEMENT SYSTEM

By
S Fyrose R170339
N Vennela R170351
M Heena R170350

Under the Guidance of

M MuniBabu

Department of Computer Science and Engineering



Rajiv Gandhi University of Knowledge Technologies (RGUKT),
R.K.Valley , Kadapa , Andhra pradesh.

as a part of
Partial fulfillment of the degree of Bachelor of
Technology in Computer Science and Engineering

Date : 20-09-2022

CERTIFICATE OF PROJECT COMPLETION

This is to certify that the report entitled **"EMPLOYEE MANAGEMENT SYSTEM"** submitted by **S.Fyrose** bearing ID No. **R170339**, **N.Vennela** bearing ID No. **R170351** and **M.Heena** bearing ID No. **R170350** in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out by them under my supervision and guidence.

The report has not been submitted previously in part or in full to this or any other University or Institution for the award of any degree or diploma.

Project Guide,	**Head of the Department,**
M. Muni Babu,	P. Harinadha,
Assistant Professor,	Assistant Professor,
Dept of CSE,	Dept of CSE,
RK.Valley, RGUKT	RK.Valley, RGUKT

Declaration

We, hereby declare that this report entitled **“Employee Management System”** submitted by us under the guidance and supervision of **Mr.M.Muni Babu** is a bonafied work. We also declare that it has not been submitted previously in part or in full to this university or other university or institution for the award of any degree or diploma.

We will be solenly responsible if any kind of plagiarism is found.

Date:16-09-2022
Place:RK Valley

R170339, S Fyrose
R170351, N Vennela
R170350, M Heena

Acknowledgement

We would like to express our sincere gratitude to **M. Muni Babu**, Our project guide for valuable suggestions and keen interest throughout the period of project work.

We are grateful to **P.Harinadha, HOD OF CSE** for providing excellent computing facilities and a congenial atmosphere for progressing with our project.

At the outset, I would like to thank **Rajiv Gandhi University of Knowledge Technologies, RK Valley** for providing all the necessary resources for the successful completion of our project work.

We express our thanks to all those who contributed for the successfull completion of our project work.

With gratitude,

S Fyrose R170339
N Vennela R170351
M Heena R170350

ABSTRACT

This report includes a development presentation of an information system for managing the staff data within a small company or organization. The system as such as it has been developed is called Employee Management System. It consists of functionally related GUI(application programm) and database. The choice of the programming tools is individual and particular.

Table of Contents

1. INTRODUCTION

1.1 Introduction to project

Employee Management System is desktop application, developed to maintain the details of employees working in any organization. It maintains the information about the personal and official details of the employees.

1.2 Importance of the Project

The main importance of the system is to computerize the maintenance of the employee details in an organization. It also includes the details about the employee personal.

To develop an well-designed database to store employee information. Provides full functional reports to management of the company.

1.3 Background

Most of the contemporary Information systems are based on the Database technology as a collection of logically related data, and DBMS as a software system allowing the users to define, create, maintain and control access to the database.The process of constructing such kind of systems is not so simple. It involves a mutual development of application program and database. The application program is actually the bridge between the users and the database, where the data is stored. Thus, the well-developed application program and database are very important for the reliabili-ty, flexibility and functionality of the system.

Then so defined systems differentiate to each other and their development comprises a great variety of tasks to be resolved and implemented.

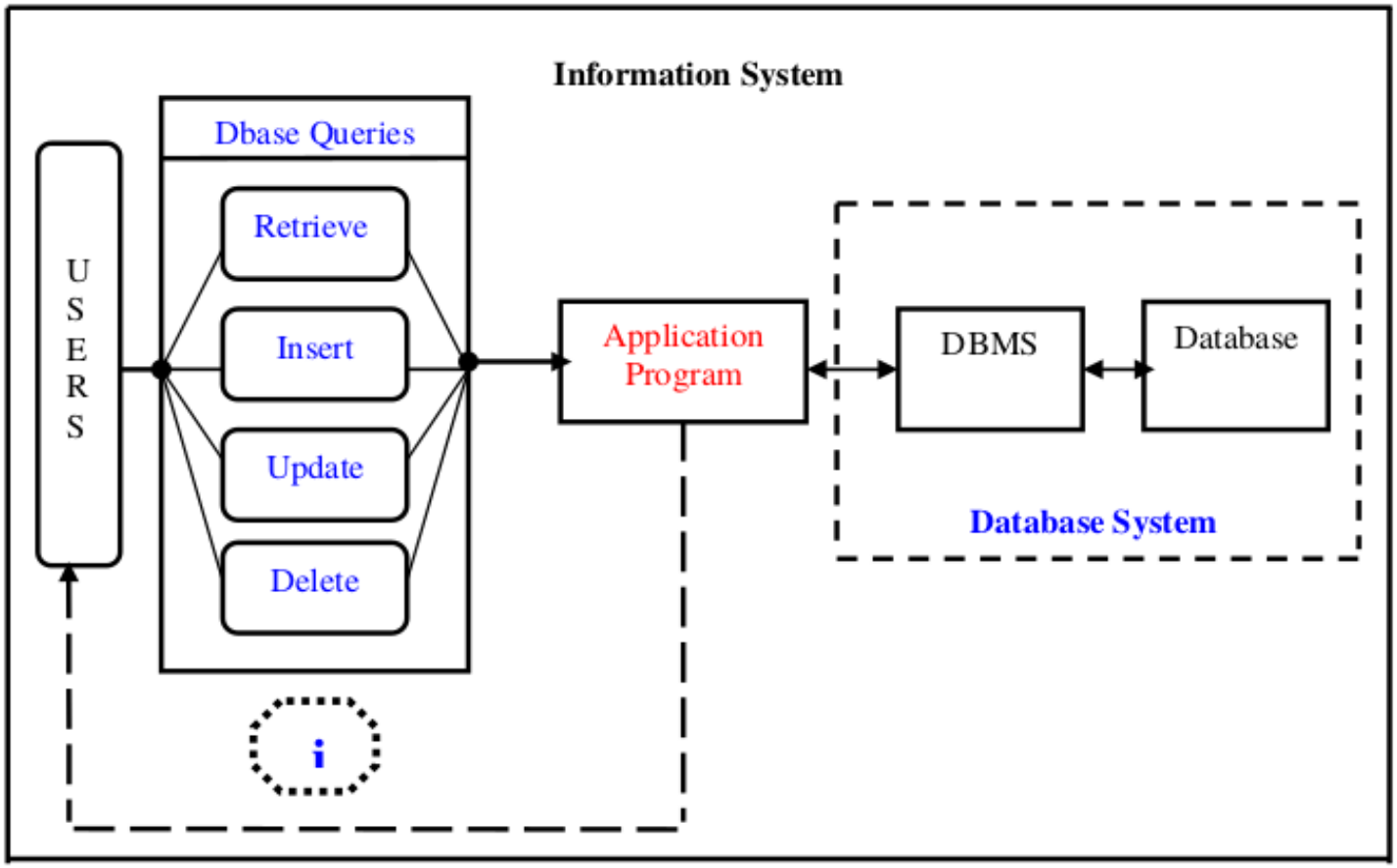


Figure 1.1 Database information systems - principle scheme

2. Structure of Project

2.1 Program's Structure Analyzing and GUI Constructing

After getting the database prepared, application program should be constructed and implemented in some programming environment to enable the users to communicate with the database.

Graphical User Interface (GUI) is intended to be built up as a basic structure of the program. The first general advice when constructing GUIs is to "know thy users" as there is a large number of rules and requirements, concerning the whole process of GUIdevelopment. Every GUI consists of certain number of controls (text-boxes, combo-boxes, buttons...etc.). The list of all properties and methods for all controls is called Application Programming Interface (API).

A set of controls is used in order to reach the desired purpose, what concerns the functionality of the application, including Labels, Text boxes, Combo Boxes, Data Grid, Buttons, Group Boxes, Panels, Tab controls etc. All of these controls, available in the program, are fitted to the corresponding forms that are used in the application.

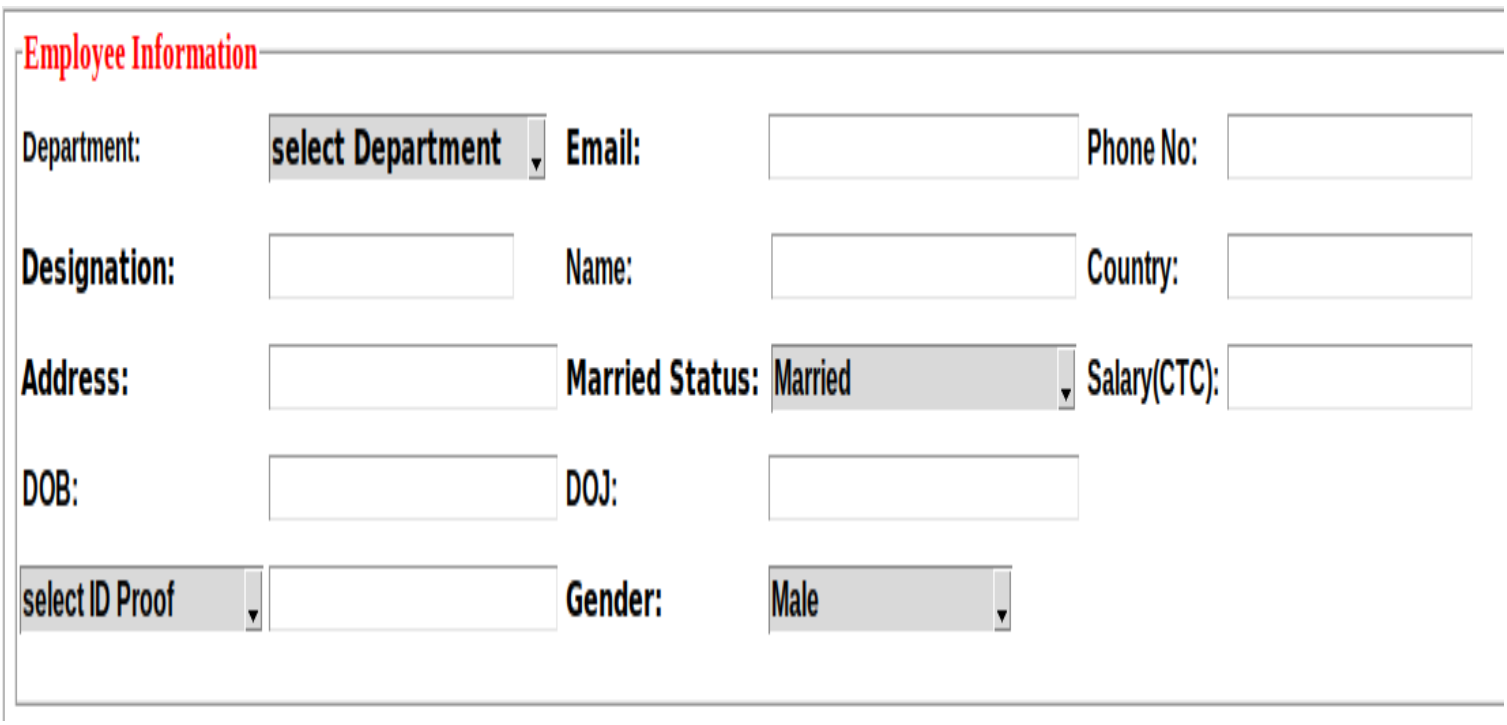


Figure 2. Frame part in GUI

In order to be able to perform different operations upon the controls and their responding data, a set of Buttons is also included in the program's implementation.

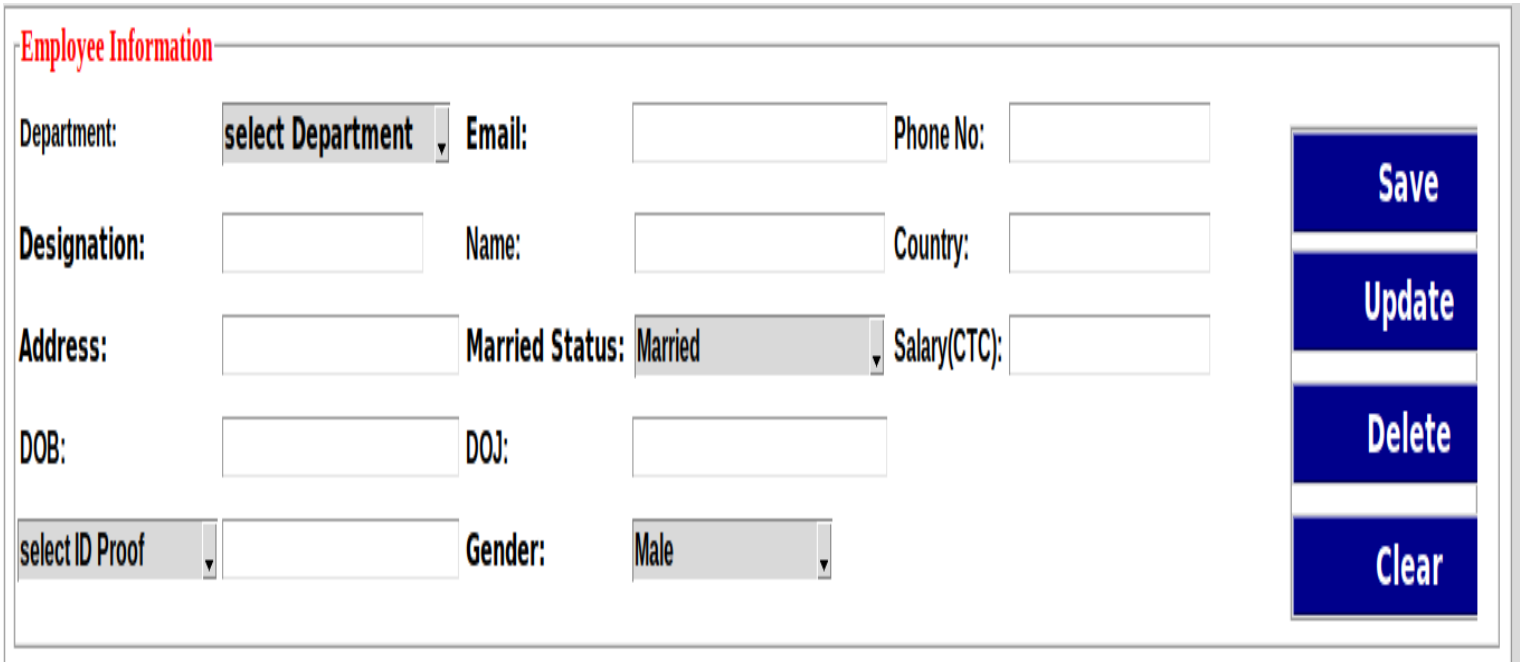


Figure 2.2 Frame Consisting of Button Frame

3. Implimentation

3.1 Database Connections and Code-Implementation

```
from tkinter import*
from tkinter import ttk
from PIL import Image, ImageTk
import mysql.connector
from tkinter import messagebox
class employee:
    def __init__(self,root):# here instead root u can use anyother words like master .it just indicates window name
        self.root=root#initialisation
        self.root.geometry("1530x790+0+0")#1530 is fro width, 790 is for height
        self.root.title('Employee Management System')
        #Variables
        self.var_dep=StringVar()
        self.var_name=StringVar()
        self.var_designation=StringVar()
        self.var_email=StringVar()
        self.var_address=StringVar()
        self.var_married=StringVar()
        self.var_dob=StringVar()
        self.var_doj=StringVar()
        self.var_idproofcombo=StringVar()
        self.var_idproof=StringVar()
        self.var_gender=StringVar()
        self.var_phone=StringVar()
        self.var_country=StringVar()
        self.var_salary=StringVar()

        lbl_title=Label(self.root,text='EMPLOYEE MANAGEMENT SYSTEM',font=('times new roman',37,'bold'),fg='darkblue',bg='white')# self.root means
on which window does this happening means named as root(window name)
       #fg=foreground for letters coloring
        lbl_title.place(x=0,y=0,width=1530,height=50)#to show this title on window
        img_logo=Image.open('collage/logo.png')
        img_logo=img_logo.resize((50,50),Image.ANTIALIAS)# ANTIALIAS is for to convert high level image to low level image
        self.photo_logo=ImageTk.PhotoImage(img_logo)# the reason to keep self. to photo_logo attribut is if u want to use this attribute anywher
within a class so u ahve to put self.(dot)
        # (above)i have installed separetely imagetk using sudo apt install python3-pip.imagetk
        #NOW I can use imagetk
        self.logo=Label(self.root,image=self.photo_logo)
        self.logo.place(x=270,y=0,width=40,height=40)
```

F igure3.1 Database connecting to Python code

3.2 Saving data into the database

This kind of operation upon the database is subdivided into two groups: Saving a new employee's records (Populating all of the tables with data) and Add a record to an employee's data records.

Saving new employee's records: The whole process comprises a few actions, but not all of them are compulsory to be accomplished at once! First of all, to unlock the fields in order to get them prepared for accepting new data, the ("Save") button has to be clicked. Afterwards, we can go to the desired form and fill the required data in. It's not necessary to fill in all of the forms with an exception of the two first, which ones hold the data for the parent table into the database, and to be able to perform a successful save into the database, we need to fill in all of the fields required there! Ofcourse, if not all of the rest forms are populated with data, a message appears on screen asking the user whether he would like to proceed anyway saving only the data, filled till the moment, or go back and fill them in.

```
def  add_data(self):
        #i want to put conditions like some validation required while entering data by user
        if self.var_dep.get()=="" or self.var_email.get()=="":#get()method is used to get data
                messagebox.showerror('Error','All field are required')# if either one is true then i have to show a message through showerror
        else:
                #if i get any error then this try block will understand how to run
                try:
                        #now we have to estabilish a connection with database
                        conn=mysql.connector.connect(host='localhost',user='root',password='iiits123',database='mydat')
                        my_cursor=conn.cursor()# need to creat a cursor
                        my_cursor.execute('insert into employee1 values(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%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_idproofcombo.get(),
                                                                                self.var_idproof.get(),
                                                                                self.var_gender.get(),
                                                                                self.var_phone.get(),
                                                                                self.var_country.get(),
                                                                                self.var_salary.get()))#this execute command is
olongs to mysql %s can be called as holding place
                        conn.commit()
                        self.fetch_data()#this is required because when we give data in middle circumstance then it should also add into
atabase
                        conn.close()
                        messagebox.showinfo('success','Employee has been added!',parent=self.root)
                except Exception as es:
                        messagebox.showerror('Error',f'Due To:{str(es)}',parent=self.root)
```

Figure 3.2 code for adding data

Below is the code for enabiling the button in order to perform above code. This makes a result over saving the data into a database.

```
        #Button_Frame
button_frame=LabelFrame(upper_frame,bd=2,relief=RIDGE,bg='white')
button_frame.place(x=1140,y=18,width=168,height=180)

btn_add=Button(button_frame,text='Save',command=self.add_data,font=("arail",15,"bold"),width=13,bg='darkblue',fg='white')
btn_add.grid(row=0,column=0,padx=0,pady=0)
```

Figure 3.3 Code for Button

After saving data :

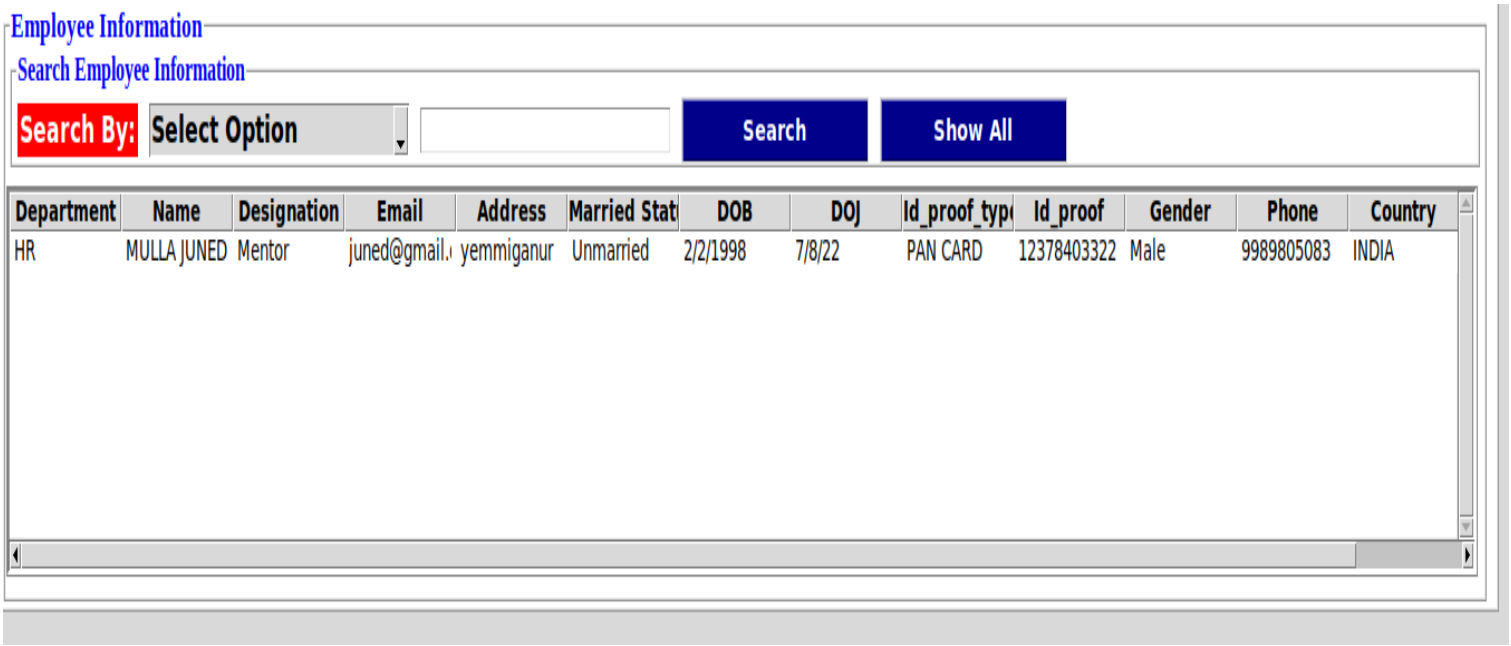


Figure 3.4 displaying data at front end

Backend:

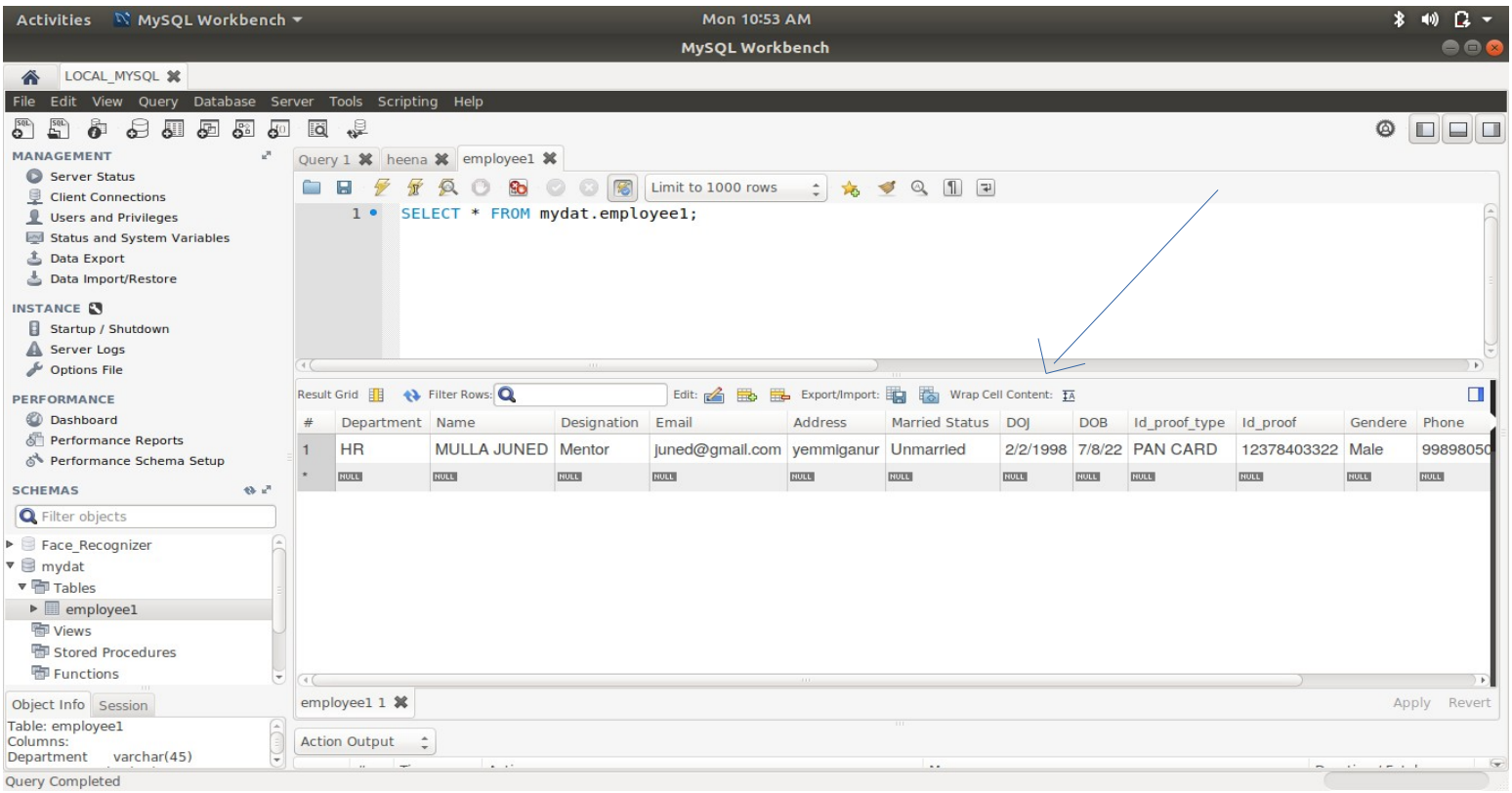


Figure 3.5 storing details at backend(WorkBench)

3.3 Code for desktop application

```
[Desktop Entry]
Name = EmployeeApp
Icon = /home/student/Desktop/Employ_Management/logo.png
Exec = python3 /home/student/Desktop/Employ_Management/test.py
Type = Application
Terminal = true
Version = 0.99

```

Figure 3.6 code for application(like stand alone)

```
#!/usr/bin/python3
from tkinter import*
from tkinter import ttk
from PIL import Image, ImageTk
import mysql.connector
from tkinter import messagebox
class employee:
    def __init__(self,root):# here instead root u can
        self.root=root#initialisation
        self.root.geometry("1530x790+0+0")#1530 is fro
        self.root.title('Employee Management System')
        #Variables
        self.var_dep=StringVar()
        self.var_name=StringVar()
        self.var_designation=StringVar()
        self.var_email=StringVar()
        self.var_address=StringVar()
        self.var_married=StringVar()
        self.var_dob=StringVar()
        self.var_doj=StringVar()
```

Figure 3.7 adding designator line to the file with extension of .py

4. RESULT

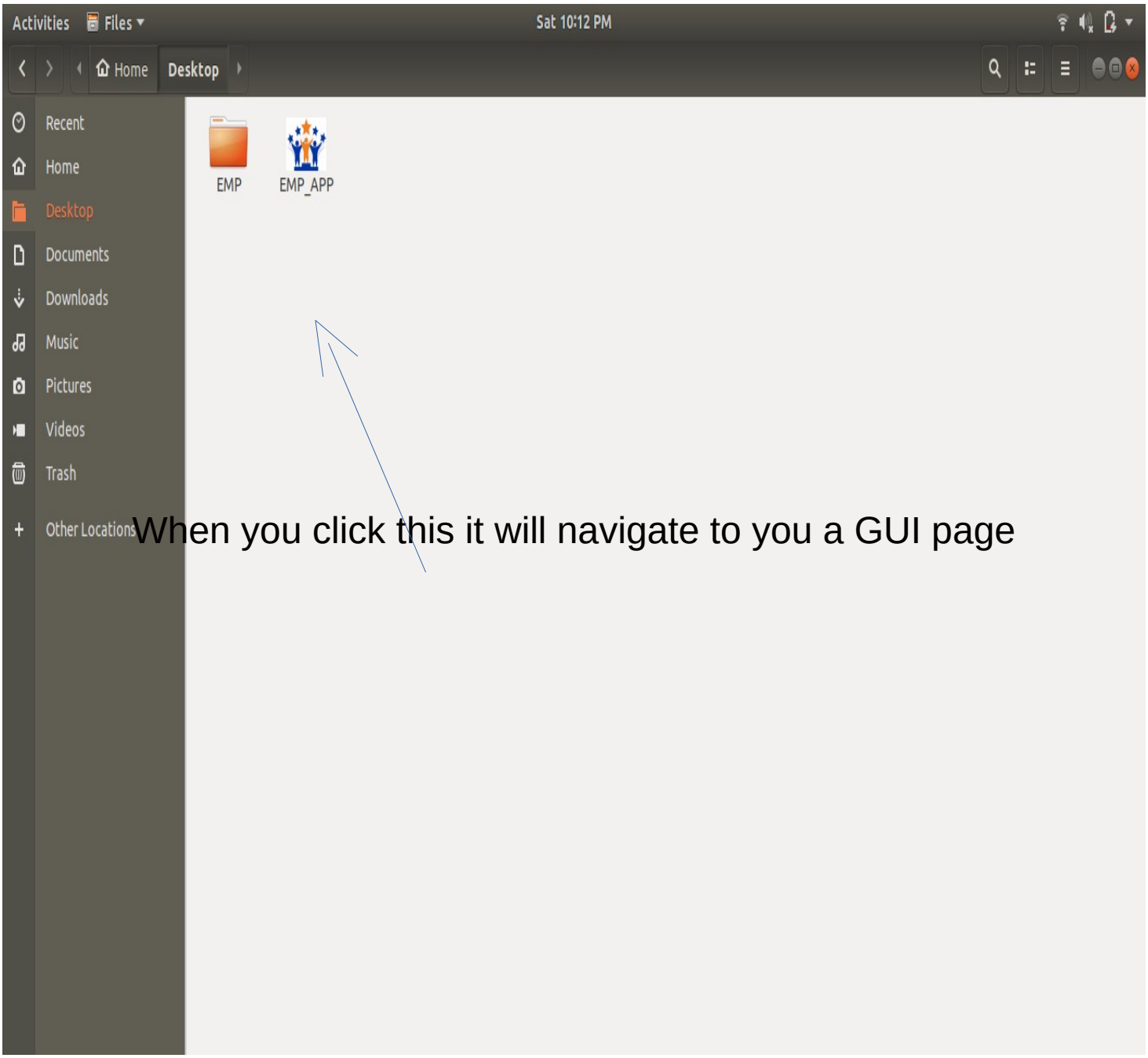


Figure 4.1 clicking desktop application(EMP_APP)

Figure 4.2 Home page of EMP

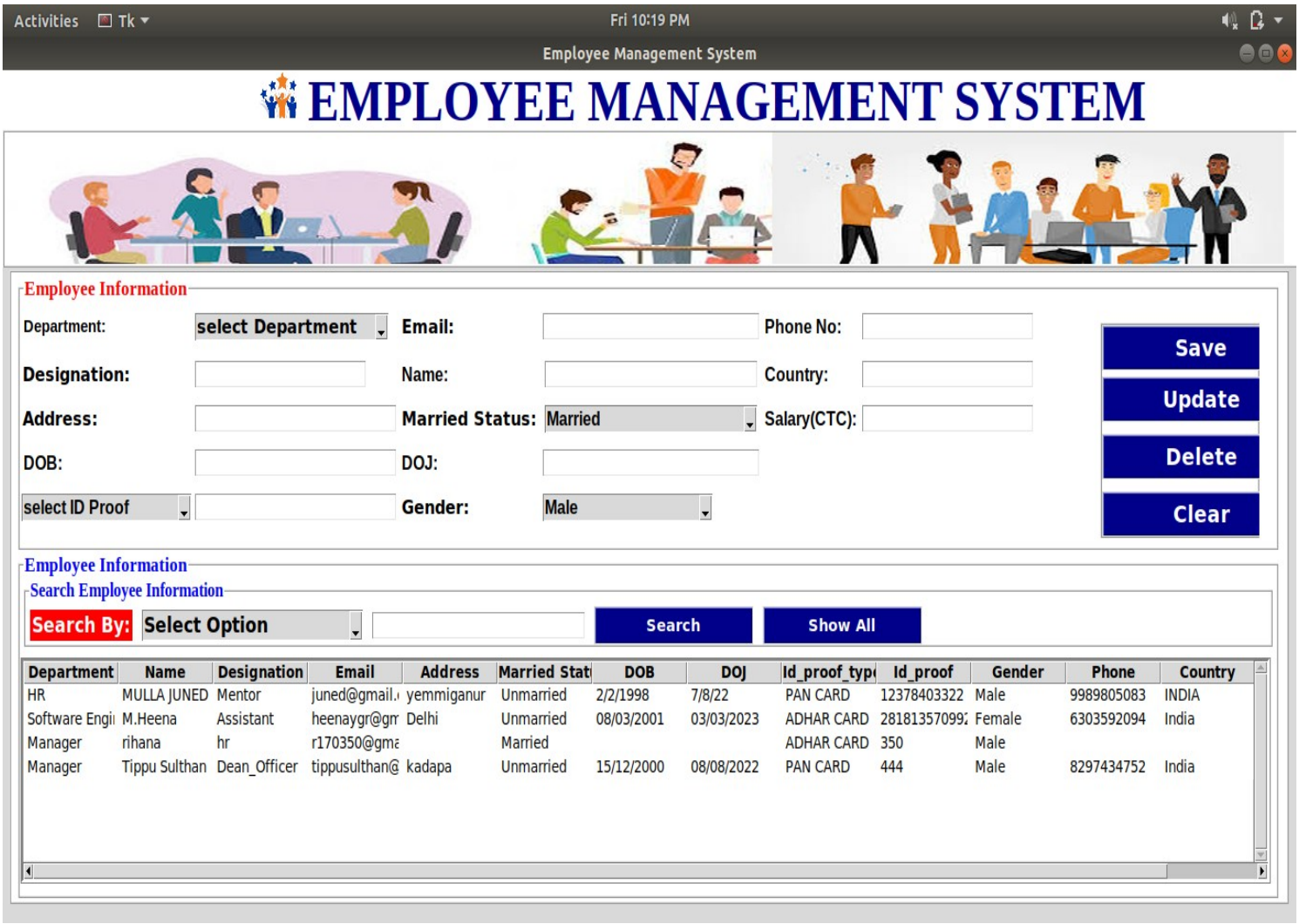


Figure 4.3 Employee Management System GUI

5. Conclusion

The employee management system helps your organization improve workforce productivity and boost overall well-being by tracking and monitoring the details of every employee. This employee management system is developed in order to computerize the activities which take more time, if done manually.

Apparently, the role of such systems is basic and essential within each company that wants to keep a really good control and record concerning its personnel data, functionality and performance on all levels in its structure. Every organization, in nowadays, has the necessity of managing its staff on a really good level as the staff has definitely the greatest merit of building up a company as such as it is.

6. Future Scope

The furture implementation of this project is adding face recognition attendance and based on this salary will be calculated and updated into details of the employees.

7. References

- Youtube
- Edureka channel
- Google
- Geeks for Geeks