**UNIVERSITY OF MUMBAI**



A DISSERTATION REPORT ON

**“EMPLOYEE RECORD MANAGEMENT SYSTEM USING PYTHON”**

SUBMITTED IN PARTIAL FULFILMENT FOR

THE REQUIREMENTS OF THE DEGREE

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER ENGINEERING**

GROUP MEMEBERS

**JAYESH BHALCHANDRA PATIL(36)**

**AMEYA ANIL RATNAPARKHI(40)**

**YASH DAMODAR SHETTIGAR(47)**

**DIPAK DASHARATH ZAD(58)**

UNDER THE GUIDANCE OF

**PROF.DIPT RAUT**



**DEPARTMENT OF COMPUTER ENGINEERING**

**G.V. ACHARYA INSTITUTE OF ENGINEERING AND TECHNOLOGY**

**UNIVERSITY OF MUMBAI**

2020-21

**DECLARATION**

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Jayesh Bhalchandra Patil(36)\_\_\_\_\_\_\_\_\_\_\_\_

Amey Anil Ratnaparkhi(40) \_\_\_\_\_\_\_\_\_\_\_\_

Yash Damodar Shettigar(47)\_\_\_\_\_\_\_\_\_\_\_\_

Dipak Dasharath Zad(58)\_\_\_\_\_\_\_\_\_\_\_\_

Date :

Place : SHELU

**PROJECT REPORT APPROVAL FOR D.S.E.**

This project report entitled **“EMPLOYEE RECORD MANAGEMENT SYSTEM USING PYTHON”** by **“Jayesh B Patil” (36), “Amey A Ratnaparkhi” (40), “Yash Damodar Shettigar” (47)**, **“Dipak Dasharath Zad” (58)** is approved for the degree of **“Bachelor of** **Computer Engineering”.**

Examiners

1.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date :

Place :SHELU

**CERTIFICATE**

This is to certify that the project entitled **“EMPLOYEE RECORD MANAGEMENT SYSTEM USING PYTHON”** is a bonafide work of “**Jayesh B Patil” (36), “Amey A Ratnaparkhi” (40),** “**Yash D Shettigar” (47), “Dipak D Zad”(58)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **“Undergraduate ”** in **“Bachelor of Computer Engineering”.**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Dipti Raut Prof. Roshan Bauskar

Project Guide Head of Department

Department of Computer Engineering Computer Engineering

G.V.A.I.E.T G.V.A.I.E.T

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Rajashri Chaudari Dr. Prashant Sonare

Project Co-ordinator Principal

Department of Computer Engineering G.V.A.I.E.T

G.V.A.I.E.T

**ACKNOWLEDGEMENT**

I take the opportunity to thank all of those who have generously helped me to give a proper shape to my work and complete my **D.S.E** project synopsis successfully. A successful project’s completion involves fruitful combination of many people, some directly involved and some indirectly, by providing support and encouragement. So with gratitude I acknowledge all those whose guidance and encouragement served a beacon of light and crowned our efforts with success.

I am thankful to Principal **Dr.Prashant Sonare**, for the constant support and encouragement during the project. It’s also a great pleasure to express my deepest gratitude to all faculty members of my department for their cooperation.

I consider it a privilege and honor to express my sincere gratitude and respect to my project guide **Prof.Dipti Raut**, Department of Computer Engineering for his valuable guidance throughout the tenure of this project seminar.

I would again like to thank **Prof.Roshan Bauskar**, HOD, Department of Computer Engineering who shared his opinions and experiences through which I received the required information which was crucial for this project.

Group Members

Jayesh B Patil (36)

Amey A Ratnaparkhi (40)

Yash D Shettigar (47)

Dipak D Zad(58)

**TABLE OF CONTENT**

**ABSTRACT**

1. INTRODUCTION
2. MOTIVATION
3. SYSTEM REQUIREMENT
4. PROBLEM STATEMENT
5. FLOWCHART
6. ALGORITHM
7. WORKING
8. IMPLEMENTATION
9. CODE
10. SNAPSHOTS
11. FEATURE OF PROJECT
12. CONCLUSION
13. REFERENCE

**ABSTRACT**

The objective of “Employee Management System” is designing a scheduling system for a work center. Scheduling is such a tool with which the process of intimating activities and notifications will be easy and even online in the organization where it is installed. But these task of scheduling the different activities if manually done whether they may be personal or official is time consuming and also may lead to confusion if not properly scheduled. Employee Management System is a distributed application, developed to maintain the details of employees working in any organization. It maintains the information about the personal details of their employees. The application is actually a suite of applications developed using Python. It is simple to understand and can be used by anyone who is not even familiar with simple employee’s 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 or organization. This software project has been developed using the powerful coding tools of HTML, CSS and PHP at Front End and Microsoft SQL Server at Back End. The software is very user friendly. The project contains modules like Employee and Admin.

**INTRODUCTION**

Employees are the backbone of any company therefore their management plays a major role in deciding the success of an organization. Employee Record System (ERS) used to manage the data of the employees such as personal details, categories etc. This system manage Employee record and data. ERS makes it easy for the employer to keep the track of all the records. This software allows the administrator to add employee, search, delete. Employee Record System is a distributed application, developed to maintain the details of employees working in any organization. The ERS has been developed to override the problems prevailing in the practicing manual system. It maintains the information about the personal and official details of the employees. This project aims to simplify the task of maintaining records of the employees of Company. To develop a well-designed database to store employee information provides full functional reports to management of Company. The objective of this project is to provide a comprehensive approach towards the management of employee information.

This Project is easy to understand and anyone can use it.  Provide computerized system for maintaining records, More efficient & reliable, Less time consuming and easy to use, Huge data storage with less computer memory, Avoid Human errors & efforts for Maintaining daily data, Avoid Data manipulations.

It has a security with Username and Password which will ensure the Protection of employee data.

**MOTIVATION**

We get our motivation from the hard working employee from any Organization \Company\ Institution. In Older times there Where any rarely use of computer where taken place. And all the records of the employees was Written on books\Cataloges or they manages Separate files for each employee and though it was very time consuming to Search each and every files for the details of employee and though there was Not only single employee working there where several workers\employees so files containing their Records was obtaining Ample of Space in the office and though it was very difficult and time consuming there Was a need for Some Solution which Saves Time and Space in the office. For Saving that time and space occupied, the Employee Management System has been discover. By using this System the user can view all the Information of Employee in Single Click. In other words user can obtain all the information on single platform. And hence it also save the Time and Space of the individual. He only has to put some Specific\Required Information by the System and by clicking on Specific option like Add\New etc. the user will get all the required Data regarding the employee on One terminal\platform. Through this the time and Space is saved User will get all Data on Single Platform by Single Click.

**SYSTEM REQUIREMENT SPECIFICATION**

**Hardware requirements:**

* 1.5 GHz Pentium 4 processor or other compatible
* 512 MB – 1 GB RAM
* Color Monitor
* Keyboard
* Mouse

**Software requirements:**

* Operating System : Microsoft Windows
* Application Software : Python 3.9.4-amd64

Microsoft Excel Worksheet

**PROBLEM STATEMENT**

Our Project Undergone a lot of changes during its development cycle, we have Fixed up all the flaws in Project which were as follows:-

* Earlier the System had major **Security Concern**, now we have implement a **Login System** which is secure and will protect the employee data.
* **Database Capacity & Securities** is improved & increased.
* **New Attributes** like blood group, address, phone no & email id therefore we’ll have more specific information of employees.
* **Search action** was just displaying the existence of entered employee but now it has gone a full rework now admin can search employee and get full information about them.
* **New GUI Update** New colors and fresh look of the Project.

**FLOWCHART**

**START**

YES

NO

NO

**IF THE ENTERED DATA IS VALID**

**ENTER THE USERNAME & PASSWORD**

**ADD AN EMPLOYEE RECORD**

**REMOVE AN EMPLOYEE RECORD**

**SELECT THE OPERATION TO PERFORM**

**DISPLAY AN ERROR MESSAGE**

**ENTER THE OTHER EMPLOYEE DETAILS**

**SEARCH AN EMPLOYEE RECORD**

**ENTER THE FIRST NAME**

**ENTER THE LAST NAME**

**DISPLAY A SUCCESS MESSAGE**

**IF THE ENTERED DATA IS VALID**

**ADD/SEARCH/REMOVE ENTERED EMPLOYEE RECORD**

**END**

**ALGORITHM**

1. Start
2. Enter the Login credentials
3. If the entered data is valid, allow access to the system
4. Select an operation to perform
5. Enter the employee details & information
6. If the entered data is valid print success message
7. else print error message and go to step 5 again
8. System will Add/Search/Remove the entered employee record
9. End

**WORKING**

**The direct functionality of Employee Record management system include:**

* Add Employee Details
* Remove Employee Details
* Search Employee Details

**Add Employee Details:**

1. Click on the Employees option located on the left side of the main dashboard

2. Then, click on the ‘Add Employee’ button, and below employee.

3. Next, you need to add the component details as per the components added to the system.

**Remove Employee Details:**

If you want to remove the created employee record from the system, you can do so by clicking on the remove option to remove the employee record ,and after clicking on this Enter employee name and surname , after entering the details next click on “click to remove “ option and the record is completely from the system.

**Search Employee Details:**

Firstly click on search option for Search Employee Details, after that Enter employee name and surname, after entering the details next click on “click to find “ option and the system show Searched employee record from the system.

**IMPLEMENTATION**

The task is to create a Database-driven Employee Record System in Python that will store the information in the Database. The idea is that we perform different changes in our Employee Record by using different functions for example the Add\_Employee will insert a new row in our Employee, also, we will create a Remove Employee Function which will delete the record of any particular existing employee in our Employee table. We can also have the information about all the existing employees by using the Display Employee function. 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 can also Search the Employee with proper details to carry out different operations in Python for input/output and for data storage this project makes the maximum utilization of the collection data type list. Python also provides varieties of data types like integer, string, Boolean, float etc. Some of the data types and data structures were used while writing the program to store and manipulate the data and perform various operate This is a simple GUI based application which is very easy to understand and use. It uses Tkinter module for the GUI. Talking about the application, the user can add, search, and remove the employee records. All the data are stored in an excel sheet. String, Boolean, List are used in this project.

**CODING**

#Employee Record System

from tkinter import\*

from tkinter import messagebox

from openpyxl import load\_workbook

import xlrd

import pandas as pd

def emp\_dict(\*args): #To add a new entry and check if entry already exist in excel sheet

#print("done")

workbook\_name="sample.xlsx"

workbook=xlrd.open\_workbook(workbook\_name)

worksheet=workbook.sheet\_by\_index(0)

wb=load\_workbook(workbook\_name)

page=wb["Employee"]

p=0

for i in range(worksheet.nrows):

for j in range(worksheet.ncols):

cellvalue=worksheet.cell\_value(i,j)

print(cellvalue)

sheet\_data.append([])

sheet\_data[p]=cellvalue

p+=1

print(sheet\_data)

fl=firstname.get()

fsl=fl.lower()

ll=lastname.get()

lsl=ll.lower()

if (fsl and lsl) in sheet\_data:

print("found")

messagebox.showerror("Error","This Employee already exist")

else:

print("not found")

for info in args:

page.append(info)

messagebox.showinfo("Done","Successfully added the employee record")

wb.save(filename=workbook\_name)

def add\_entries(): #to append all data and add entries on click the button

a=" "

e=empid.get()

ei=e.lower()

f=firstname.get()

f1=f.lower()

l=lastname.get()

l1=l.lower()

d=dept.get()

d1=d.lower()

de=designation.get()

de1=de.lower()

ad=empaddress.get()

ea=ad.lower()

pno=emppno.get()

epn=pno.lower()

blg=empbg.get()

ebg=blg.lower()

ml=empmail.get()

em=ml.lower()

list1=list(a)

list1.append(ei)

list1.append(f1)

list1.append(l1)

list1.append(d1)

list1.append(de1)

list1.append(ea)

list1.append(epn)

list1.append(ebg)

list1.append(ml)

emp\_dict(list1)

print(list1)

def add\_info(): #for taking user input to add the enteries

frame2.pack\_forget()

frame3.pack\_forget()

emp\_id=Label(frame1,text="Enter employee ID: ",bg="red",fg="white",font=('Lato',11,'bold'))

emp\_id.grid(row=1,column=1,padx=10)

e0=Entry(frame1,textvariable=empid)

e0.grid(row=1,column=2,padx=10)

emp\_first\_name=Label(frame1,text="Enter first name of the employee: ",bg="red",fg="white",font=('Lato',11,'bold'))

emp\_first\_name.grid(row=2,column=1,padx=10)

e1=Entry(frame1,textvariable=firstname)

e1.grid(row=2,column=2,padx=10)

e1.focus()

emp\_last\_name=Label(frame1,text="Enter last name of the employee: ",bg="red",fg="white",font=('Lato',11,'bold'))

emp\_last\_name.grid(row=3,column=1,padx=10)

e2=Entry(frame1,textvariable=lastname)

e2.grid(row=3,column=2,padx=10)

emp\_dept=Label(frame1,text="Select department of employee: ",bg="red",fg="white",font=('Lato',11,'bold'))

emp\_dept.grid(row=4,column=1,padx=10)

dept.set("Select Option")

e4=OptionMenu(frame1,dept,"Select Option","IT","Operations","Sales")

e4.configure(font=('Rubik',10,'bold'), fg='white', bg='black', border='0')

e4.grid(row=4,column=2,padx=10)

emp\_desig=Label(frame1,text="Select designation of Employee: ",bg="red",fg="white",font=('Lato',11,'bold'))

emp\_desig.grid(row=5,column=1,padx=10)

designation.set("Select Option")

e5=OptionMenu(frame1,designation,"Select Option","Manager","Asst Manager","Project Manager","Team Lead","Senior Tester",

"Junior Tester","Senior Developer","Junior Developer","Intern")

e5.configure(font=('Rubik',10,'bold'), fg='white', bg='black', border='0')

e5.grid(row=5,column=2,padx=10)

emp\_address=Label(frame1,text="Enter employee address: ",bg="red",fg="white",font=('Lato',11,'bold'))

emp\_address.grid(row=6,column=1,padx=10)

e6=Entry(frame1,textvariable=empaddress)

e6.grid(row=6,column=2,padx=10)

emp\_phone\_number=Label(frame1,text="Enter employee phone no: ",bg="red",fg="white",font=('Lato',11,'bold'))

emp\_phone\_number.grid(row=7,column=1,padx=10)

e7=Entry(frame1,textvariable=emppno)

e7.grid(row=7,column=2,padx=10)

emp\_blood\_group=Label(frame1,text="Enter employee blood grp: ",bg="red",fg="white",font=('Lato',11,'bold'))

emp\_blood\_group.grid(row=8,column=1,padx=10)

e8=Entry(frame1,textvariable=empbg)

e8.grid(row=8,column=2,padx=10)

emp\_mail=Label(frame1,text="Enter employee mail id: ",bg="red",fg="white",font=('Lato',11,'bold'))

emp\_mail.grid(row=9,column=1,padx=10)

e9=Entry(frame1,textvariable=empmail)

e9.grid(row=9,column=2,padx=10)

button4=Button(frame1,text="Add Employee",command=add\_entries,font=('Rubik',10,'bold'), fg='white', bg='black')

button4.grid(row=10,column=2,pady=10)

frame1.configure(background="Red")

frame1.pack(pady=10)

def clear\_all():

# f.pack\_forget() #for clearing the entry widgets

frame1.pack\_forget()

frame2.pack\_forget()

frame3.pack\_forget()

loginF.pack\_forget()

def remove\_emp(): #for taking user input to remove enteries

clear\_all()

emp\_first\_name=Label(frame2,text="Enter first name of the employee:",bg="red",fg="white",font=('Lato',11,'bold'))

emp\_first\_name.grid(row=1,column=1,padx=10)

e10=Entry(frame2,textvariable=remove\_firstname)

e10.grid(row=1,column=2,padx=10)

e10.focus()

emp\_last\_name=Label(frame2,text="Enter last name of the employee:",bg="red",fg="white",font=('Lato',11,'bold'))

emp\_last\_name.grid(row=2,column=1,padx=10)

e11=Entry(frame2,textvariable=remove\_lastname)

e11.grid(row=2,column=2,padx=10)

remove\_button=Button(frame2,text="Remove Employee",command=remove\_entry,font=('Rubik',10,'bold'), fg='white', bg='black')

remove\_button.grid(row=3,column=2,pady=10)

frame2.configure(background="Red")

frame2.pack(pady=10)

def remove\_entry(): #to remove entry from excel sheet

rsf=remove\_firstname.get()

rsf1=rsf.lower()

print(rsf1)

rsl=remove\_lastname.get()

rsl1=rsl.lower()

print(rsl1)

# workbook\_name="sample.xlsx"

path="sample.xlsx"

wb = xlrd.open\_workbook(path)

sheet = wb.sheet\_by\_index(0)

for row\_num in range(sheet.nrows):

row\_value = sheet.row\_values(row\_num)

print(row\_value)

if (row\_value[2]==rsf1 and row\_value[3]==rsl1):

print(row\_value)

print("found")

file="sample.xlsx"

x=pd.ExcelFile(file)

writer=pd.ExcelWriter('sample.xlsx', engine='xlsxwriter')

df1=x.parse(x.sheet\_names[0])

df2=x.parse(x.sheet\_names[1])

df1=df1[df1['First Name']!=rsf]

dfs = {'Employee':df1,'Login':df2}

for sheet\_name in dfs.keys():

dfs[sheet\_name].to\_excel(writer,sheet\_name=sheet\_name,index=False)

writer.save()

messagebox.showinfo("Done","Successfully removed the Employee record")

break

# else:

# print("Error occured")

# messagebox.showerror("Error","Employee does not exist")

clear\_all()

def search\_emp(): #can implement search by 1st name,last name,emp id, designation

clear\_all()

emp\_first\_name=Label(frame3,text="Enter first name of the employee:",bg="red",fg="white",font=('Lato',12,'bold')) #to take user input to seach

emp\_first\_name.grid(row=1,column=1,padx=10)#place(x=10, y=20)

e12=Entry(frame3,textvariable=searchfirstname)

e12.grid(row=1,column=2,padx=10)

e12.focus()

emp\_last\_name=Label(frame3,text="Enter last name of the employee:",bg="red",fg="white",font=('Lato',12,'bold'))

emp\_last\_name.grid(row=2,column=1,padx=10)

e13=Entry(frame3,textvariable=searchlastname)

e13.grid(row=2,column=2,padx=10)

search\_button=Button(frame3,text="Search Employee",command=search\_entry,font=('Rubik',10,'bold'), fg='white', bg='black')

search\_button.grid(row=3,column=2,pady=10)

nameval=""

frame3.configure(background="Red")

frame3.pack(pady=10)

def displaySearchItem(fnameval,lnameval,deptval,designnationval,addval,phoneval,bgroupval,mailval):

searchresultF=Tk()

searchresultF.geometry("400x400")

searchresultF.title("Employee Details")

searchresultF.config(bg='red')

sresultTitle = Label(searchresultF,text="Search Result")

sresultTitle.config(font=("Helvatica",20,"bold"))

sresultTitle.grid(row=1,column=0,columnspan=6,pady=10)

slabel = Label(searchresultF,text="Employee First Name : ")

slabel.config(font=("Helvatica",10,"bold"), fg='black', bg='red')

slabel.grid(row=2,column=0,pady=10)

sfname = Label(searchresultF,text=fnameval)

sfname.config(font=("Helvatica",10,"bold"), fg='white', bg='black')

sfname.grid(row=2,column=3,pady=10)

slabel2 = Label(searchresultF,text="Employee Last Name: ")

slabel2.config(font=("Helvatica",10,"bold"), fg='black', bg='red')

slabel2.grid(row=3,column=0,pady=10)

slname = Label(searchresultF,text=lnameval)

slname.config(font=("Helvatica",10,"bold"), fg='white', bg='black')

slname.grid(row=3,column=3,pady=10)

slabel3 = Label(searchresultF,text="Employee Department : ")

slabel3.config(font=("Helvatica",10,"bold"), fg='black', bg='red')

slabel3.grid(row=4,column=0,pady=10)

sdept = Label(searchresultF,text=deptval)

sdept.config(font=("Helvatica",10,"bold"), fg='white', bg='black')

sdept.grid(row=4,column=3,pady=10)

slabel4 = Label(searchresultF,text="Employee Designation : ")

slabel4.config(font=("Helvatica",10,"bold"), fg='black', bg='red')

slabel4.grid(row=5,column=0,pady=10)

sdesignation = Label(searchresultF,text=designnationval)

sdesignation.config(font=("Helvatica",10,"bold"), fg='white', bg='black')

sdesignation.grid(row=5,column=3,pady=10)

slabel5 = Label(searchresultF,text="Employee Address : ")

slabel5.config(font=("Helvatica",10,"bold"), fg='black', bg='red')

slabel5.grid(row=6,column=0,pady=10)

sadd = Label(searchresultF,text=addval)

sadd.config(font=("Helvatica",10,"bold"), fg='white', bg='black')

sadd.grid(row=6,column=3,pady=10)

slabel6 = Label(searchresultF,text="Employee Phone Number : ")

slabel6.config(font=("Helvatica",10,"bold"), fg='black', bg='red')

slabel6.grid(row=7,column=0,pady=10)

sphone = Label(searchresultF,text=phoneval)

sphone.config(font=("Helvatica",10,"bold"), fg='white', bg='black')

sphone.grid(row=7,column=3,pady=10)

slabel7 = Label(searchresultF,text="Employee Blood Group : ")

slabel7.config(font=("Helvatica",10,"bold"), fg='black', bg='red')

slabel7.grid(row=8,column=0,pady=10)

sbgroup = Label(searchresultF,text=bgroupval)

sbgroup.config(font=("Helvatica",10,"bold"), fg='white', bg='black')

sbgroup.grid(row=8,column=3,pady=10)

slabel8 = Label(searchresultF,text="Employee Gmail : ")

slabel8.config(font=("Helvatica",10,"bold"), fg='black', bg='red')

slabel8.grid(row=9,column=0,pady=10)

smail = Label(searchresultF,text=mailval)

smail.config(font=("Helvatica",10,"bold"), fg='white', bg='black')

smail.grid(row=9,column=3,pady=10)

searchresultF.mainloop()

def search\_entry():

sf=searchfirstname.get()

ssf1=sf.lower()

print(ssf1)

sl=searchlastname.get()

ssl1=sl.lower()

print(ssl1)

path="sample.xlsx"

wb = xlrd.open\_workbook(path)

sheet = wb.sheet\_by\_index(0)

val=0

log=1

for row\_num in range(sheet.nrows):

row\_value = sheet.row\_values(row\_num)

if (row\_value[2]==ssf1 and row\_value[3]==ssl1):

fnameval=row\_value[2]

lnameval=row\_value[3]

deptval=row\_value[4]

designnationval=row\_value[5]

addval=row\_value[6]

phoneval=row\_value[7]

bgroupval=row\_value[8]

mailval=row\_value[9]

print(row\_value)

print("found")

displaySearchItem(fnameval,lnameval,deptval,designnationval,addval,phoneval,bgroupval,mailval)

clear\_all()

val=1

home()

break

else:

if(row\_value[1]!=ssf1 and row\_value[2]!=ssl1):

print("Not found")

log=0

clear\_all()

if(log==0):

if(val==0):

messagebox.showerror("Sorry","Employee Record does not Exist")

clear\_all()

frame3.pack()

def home():

clear\_all()

label2=Label(f,text="Select an action: ", background="Black", fg="White", font=('Lato',11,'bold'))

label2.pack(side=LEFT,pady=10)

button1=Button(f,text="Add", background="Red", fg="White", command=add\_info, width=8, font=('Italic',11,'bold'))

button1.pack(side=LEFT,ipadx=20,pady=10)

button2=Button(f,text="Remove", background="Red", fg="white", command=remove\_emp, width=8, font=('Italic',11,'bold'))

button2.pack(side=LEFT,ipadx=20,pady=10)

button3=Button(f,text="Search", background="Red", fg="White", command=search\_emp, width=8, font=('Italic',11,'bold'))

button3.pack(side=LEFT,ipadx=20,pady=10)

button6=Button(f,text="Close", background="Red", fg="White", width=8, command=root.destroy, font=('Italic',11,'bold'))

button6.pack(side=LEFT,ipadx=20,pady=10)

f.configure(background="Black")

f.pack()

def validatelogin():

sf=userval.get()

ssf1=sf.lower()

print(ssf1)

sl=passval.get()

ssl1=sl.lower()

print(ssl1)

path="sample.xlsx"

wb = xlrd.open\_workbook(path)

sheet = wb.sheet\_by\_index(1)

for row\_num in range(sheet.nrows):

row\_value = sheet.row\_values(row\_num)

if (row\_value[1]==ssf1 and row\_value[2]==ssl1):

clear\_all()

val=1

home()

break

else:

if(row\_value[1]!=ssf1 and row\_value[2]!=ssl1):

print("Not found")

log=0

clear\_all()

loginF.pack()

if(log==0):

if(val==0):

userval.delete(0,END)

passval.delete(0,END)

messagebox.showerror("Sorry","Username/Password Invalid")

def login():

clear\_all()

logtitle=Label(loginF,text="Login To Continue")

logtitle.config(font=('Lato',16,'bold'))

logtitle.configure(background="red")

#root.wm\_attributes('-transparentcolor',root['bg'])

# logtitle.grid(row=0,column=0,columnspan=3)

logtitle.pack()

userlabel=Label(loginF,text="Enter Username")

userlabel.config(font=('Lato',11,'bold'))

userlabel.configure(background="darkgrey")

userlabel.pack()

# userlabel.grid(row=2,column=0,columnspan=2)

global userval

userval=Entry(loginF)

userval.pack()

# userval.grid(row=2,column=0)

passlabel=Label(loginF,text="Enter Password")

passlabel.config(font=('Lato',11,'bold'))

passlabel.configure(background="darkgrey")

passlabel.pack()

# passlabel.grid(row=3,column=0,columnspan=2)

global passval

passval=Entry(loginF, show='\*')

passval.pack()

# passval.grid(row=2,column=0)

loginB=Button(loginF,text="Login", command=validatelogin, font=('Rubik',11,'bold'), fg='white')

# loginB.grid(row=4,column=2)

loginB.configure(background="black")

loginB.pack()

loginF.configure(background="darkgrey")

loginF.pack()

root=Tk()

loginF=Frame(root) #Main window

f=Frame(root)

frame1=Frame(root)

frame2=Frame(root)

frame3=Frame(root)

root.title("Employee Record Management System")

root.geometry("830x395")

#root.configure(background="ERSbg1.png")

bg = PhotoImage(file="E:\Downloads\proj\mainbg.png")

my\_label = Label (root, image=bg)

my\_label.place(x=0, y=0, relwidth=1, relheight=1)

my\_label.lower()

# scrollbar=Scrollbar(root)

# scrollbar.pack(side=RIGHT, fill=Y)

empid=StringVar() #Declaration of all variables

firstname=StringVar()

lastname=StringVar()

id=StringVar()

dept=StringVar()

designation=StringVar()

empaddress=StringVar()

emppno=StringVar()

empbg=StringVar()

empmail=StringVar()

remove\_firstname=StringVar()

remove\_lastname=StringVar()

searchfirstname=StringVar()

searchlastname=StringVar()

sheet\_data=[]

row\_data=[]

val=0

log=1

#Main window buttons and labels

label1=Label(root,text="EMPLOYEE RECORD MANAGEMENT SYSTEM")

label1.config(font=('Italic',16,'bold'), justify=CENTER, background="Black",fg="Red", anchor="center")

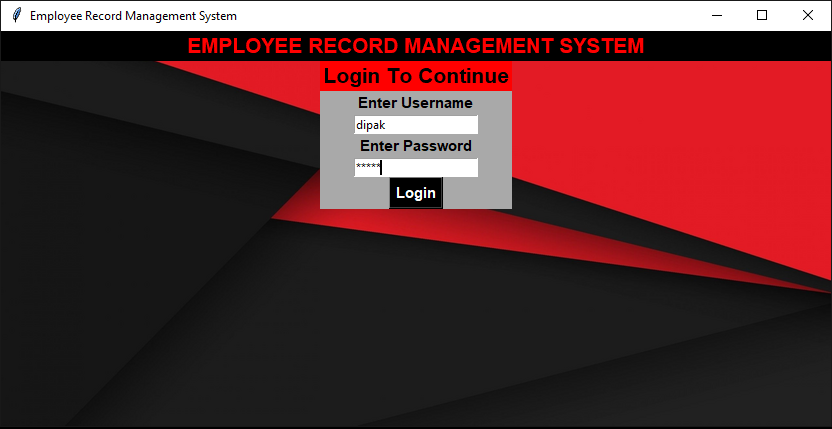
label1.pack(fill=X)

login()

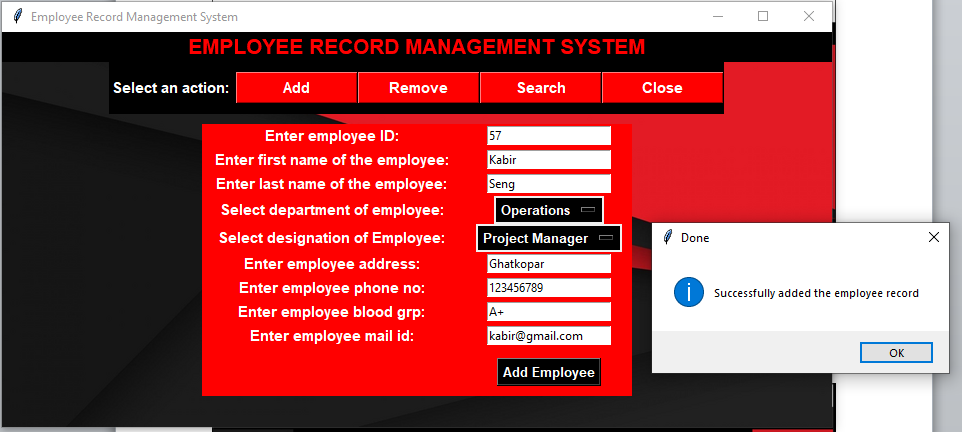
root.mainloop()

**SNAPSHOTS**

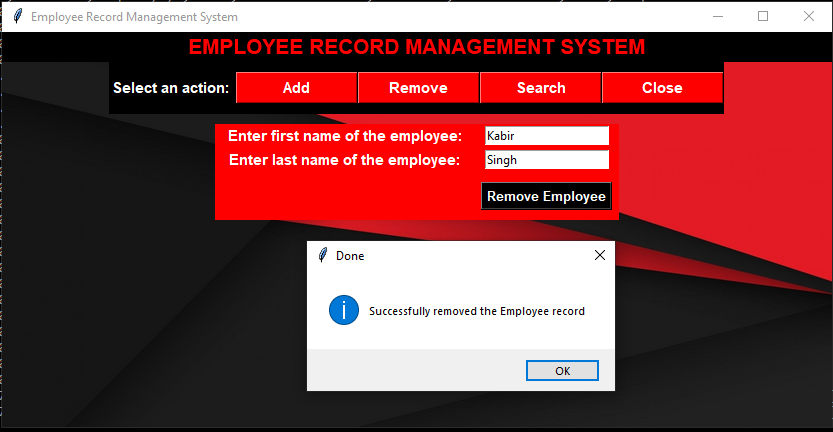
**Login page of System:**



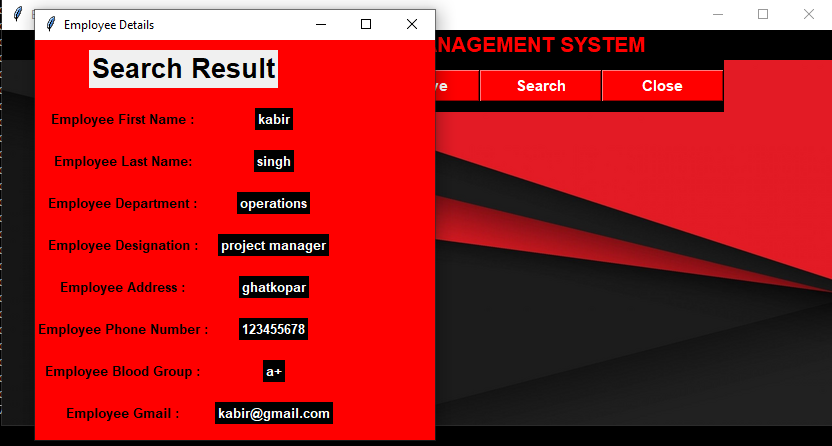
**Adding employee record in database:**



**Removing employee record from database:**



**Searching employee record from database:**



**FEATURE OF PROJECT**

* User-Friendly (as user can easily use Employee Record Management System as it has **simple GUI** created using Tkinter).
* Requires less effort for operating data as it is entered through keyboard manually & is cross verified also therefore **more accurate**.
* **Unlimited life-cycle**(system remains available as long as code remains in the computer system)
* **Requires less computing storage** & can run in low spec computer system.
* It is **fast &** **less time consuming** with computations and can manipulate data quickly into the database.
* Future proof since it has now database which is secure and password protected
* **Login Required** to get access to the database, or perform operations on database i.e add, search, delete.

**CONCLUSION**

In this report, an Employee Record Management System’s development has been presented. It Was emphasis on the basic Steps, Consequently taken during the project’s development as a particular attention was turn to the basic operative functions performed upon the data into the database.

We have developed this system, to show all the Employee data like Name, designation, etc by Single click on one platform. This system collects the data and Shows or reflects all the data on one platform. And in this system user can add\edit\remove the data of the employee by simple steps. And in this system user can add the data of the new interns or joiners.

Apparently, the role of such system is basic and essential within each company that wants to keep a really good control and record concerning its personnel data, functionally and performance of all levels in its structure. Every organization, 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. The well managed Staff means giving appropriate financial award-ness of and all kind of benefits as such as they have been deserved. That’s why the development of such system is not a programming business a lot of people are ordinarily involved in such project and one of the basic requirements is the reliability of the system, especially what concern the Storage of data and all of the operation will be performed on it.

**REFERENCES**

[www.google.com](http://www.google.com)

[www.youtube.com](http://www.youtube.com)

<http://stackoverflow.com/>

[www.w3schools.com](http://www.w3schools.com)

<http://www.codeproject.com>

<https://www.freecodecamp.org/>

<https://en.wikipedia.org/wiki/>

And some blogs