A PROJECT REPORT
ON
EMPLOYEE RECRUITMENT SYSTEM

Submitted by

Rishabh kr. Singh [11904301]
Abhay Singh [11904269]
Shubham Verma [11905678]

Under the supervision of

Mr. Moin Hassan

At

Department of Computer Science and Engineering (2019-23)

Jalandhar - Delhi, Grand Trunk Rd,
Phagwara, Punjab 144001

DATE OF SUBMISSION: 31^{ST} OCT,2020

Table of Contents

ACKNOWLEDGEMENT

We would like to thank my professor, Mr. Moin Hassan for his guidance and comments on various aspects of this project and for thoughtful contribution in our efforts

We the team working on the project of "Employee Recruitment System" are overwhelmed in all humbleness and gratefulness to acknowledge our depth to all those who have helped us to put our ideas and assigned work, well above the level of simplicity and into something concrete.

We all thank whole heartedly Moin Hassan Sir for selecting us a part of his valuable project, constantly motivating for doing better and showing complete confidence in our work.

Also, In the process, we learnt a lot other technical and non-technical things from them.

Finally, We also like to thank all other colleagues working in different projects under Moin Hassan Sir for helping us at small problems as well as critical junctures.

Other known and unknown persons who helped me in my work also deserve thanks for their co-operation. Last but not the least; I am very thankful to my friends for providing me a valuable assistance during my work.

With Pleasure

Rishabh kr. Singh

Abhay Singh

Shubham Verma

ABSTRACT

This project *Employee Recruitment System (ERS)* is a job portal system in which jobseekers can register themselves online.

Employee Recruitment System provides online help to the users all over the world. This kind of system plays an important role in simplifying the recruitment process.

It also makes it possible for organization to post their staffing requirements and view profiles of interested candidates.

Earlier recruitment was done manually and it was all at a time consuming work. Now it is all possible in a fraction of second. The system has been designed to do a whole lot more than just reduce paperwork. It can make a significant contribution to a company's marketing and sales activities. Employee recruitment system make possible for managers to access information that is crucial to managing their staff, which they can use for human resources management, staffing and planning activities. The primary purpose to develop this system is to optimize the recruitment process for an organization. Besides, the qualified applicants could be sort by this system based on their qualifications and company requirements.

Project Summary

Employee Recruitment System by overall is aims to facilitate the applicant to apply for the job online. Indirectly, it is also to facilitate the managerial department of an organization for an optimized and systematic employee recruitment process.

Online recruiting systems, with its emphasis on a more strategic decision making process is fast gaining ground as a popular outsourced function.

The system also provides the global platform for both - jobseekers and the organization, where the jobseekers can find their dream jobs and organization can find the right candidate to fulfill staff requirements.

This system has web-forms like registration form, login form and account pages like user applicant page, admin account page etc. It includes following main modules:-

- Admin module
- Candidate module

An administrator can be a manager who has full authority over the whole system. The administrator is able to update and retrieve data from the account of candidate. The candidate is a center of this system. He has to register himself to use the services of the system.

1.1 Purpose:

The primary purpose to develop this system is to optimize the recruitment process for an organization. Besides, the qualified applicants could be sort by this system based on their qualifications and company requirements.

The system has been designed to do a whole lot more than just reduce paperwork. It can make a significant contribution to a company’s marketing and sales activities

1.2 Project Scope:

Online Recruitment System enables the users to have the typical recruitment facilities and features at their disposal. It resolves typical issues of manual staffing processes and activities into a controlled and closely monitored work flow in the architecture of the application. This multi platform solution brings in by default, the basic intelligence and immense possibilities for further extension of the application as required by the user. The system makes it simpler to share and manage the organization's human resource requirements with higher efficiency and easiness. The objective of these websites is to serve as a common meeting ground for jobseekers and organization, both locally and globally. This kind of systems is specifically designed for organization to help in solving staffing problems and managing human resource department activities at high degree of optimization.

1.3 Objectives:

- This software helps applicants to find suitable job within the organization and apply for that job easily.
- The software helps in managing and viewing details of interested applicants for the administrator.
- The system is capable of sorting and filtering best suitable candidates based on some criteria.
- Company will not have to waste his time for finding right employee at right post

TECHNOLOGY AND LITERATURE REVIEW

Introduction to MySQL:

MySQL is the world's second most widely used open-source relational database management system (RDBMS). The SQL phrase stands for Structured Query Language.

The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements.

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other 'AMP' stacks).

Introduction to Tkinter:

Tkinter is an open source, portable graphical user interface (GUI) library designed for use in Python scripts. Tkinter relies on the Tk library, the GUI library used by Tcl/Tk and Perl, which is in turn implemented in C. Thus, Tkinter is implemented using multiple layers.

Features:

1. ***Layered design*****:** The layered approach used in designing Tkinter gives Tkinter all of the advantages of the TK library. Therefore, at the time of creation, Tkinter inherited from the benefits of a GUI toolkit that had been given time to mature.

2. ***Accessibility Learning****:* Tkinter is very intuitive, and therefore quick and painless. The Tkinter implementation hides the detailed and complicated calls in simple, intuitive methods. This is a continuation of the Python way of thinking, since the language excels at quickly building prototypes

MILESTONE AND DELIVERABLES

Milestones and Deliverables are the part of the Project Scheduling. In what time your Project is going to be ready, is known by Milestones. Milestone is an endpoint of the software process activity.

Software Project Activity	**Milestone**
Project Plan	Project schedule
Requirement Collection	User requirements
Data flow analysis	System Flow
Design 1. Database Design 2. GUI	System Design Document
Implementation 1. Code for giving security 2. Code for reports	Access rights Reports
Testing	Setting validations and error messages

MILESTONE AND DELIVERABLES

USER CHARACTERISTICS:

1. ADMINISTRATOR:

Admin can update, delete, modify the detail of the candidates which are filled by them only of their respective department..He also can schedule examination activities and sort candidates basis on exam results.

2. JOBSEEKER:

Jobseeker can register himself, upload CVs, find the appropriate job within organization's vacancy constraints, attend the exam and give feedback about the system.

3.1 HARDWARE AND SOFTWARE REQUIREMENT SPECIFICATION

Server Side:

Hardware

- Processor : Intel core processor 2 GHz
- RAM : 2 GB RAM
- Hard Disk : 80 GB HDD
- Monitor : Compatible Printing Device
- Keyboard : Any Keyboard
-

Software

- Operating System: Microsoft Windows 7/8/8.1/XP/VISTA
- Database: MY SQL
- Browser: Mozila Firefox or Google Chrome

Client side:

Hardware

- Processor : Intel core i3 or higher processor 2 GHz
- RAM : 520 MB RAM
- Hard Disk : 40 GB HDD
- Monitor : Compatible Printing Device
- Keyboard : Any Keyboard

Software

- MYSQL Server
- Browser: Mozila Firefox or Google Chrome
- Operating System: Microsoft Windows 7/8/8.1/XP/VISTA

CONSTRAINTS:

Constraints are basically the limitations of a software and hardware around which all thing revolves.

Hardware Limitations

The limitation of dream viewer is that it requires RAM that cannot be less than 520 MB and the processor cannot be less than 2 GHz speed as recommended in the hardware requirements.

Reliability Requirements

The main reliability requirement is the validation used. Without proper validation the system does not allow to enter that value into database. All the required validation controls are kept controls are kept to keep the system secure.

The following are the some of the reliability requirements:

- In the email ID the user cannot enter any dummy value, the validation checks that whether there is a '@' or '_' symbol in that.
- Any null value is not allowed in place of compulsory fields.
- In numeric field user cannot enter any character value.
- In date of birth, user cannot enter date and time other than given format
- Entered password and confirm password must match to each other.
- User can not re –register an account on his primary email.

Assumption and Dependencies:

- End user is the person having enough knowledge for the project operation.
- Only admin have all the privileges.
- Candidate can only fill the registration form.
- Candidate is not allowed to fetch other employee details.

REQUIREMENTS OF A NEW SYSTEM:

Functional requirements:

- The system should record all the details of an applicant.
- The system should provide applicant to edit his profile details.
- The system should allow admin to have full authority over user accounts.
- The system should allow admin to sort and filter applicants based on some criteria.
- The system should allow admin to send notifications about upcoming events and deadlines to applicants.
- Username & password are sent to the users via mobile sms and email after registration.

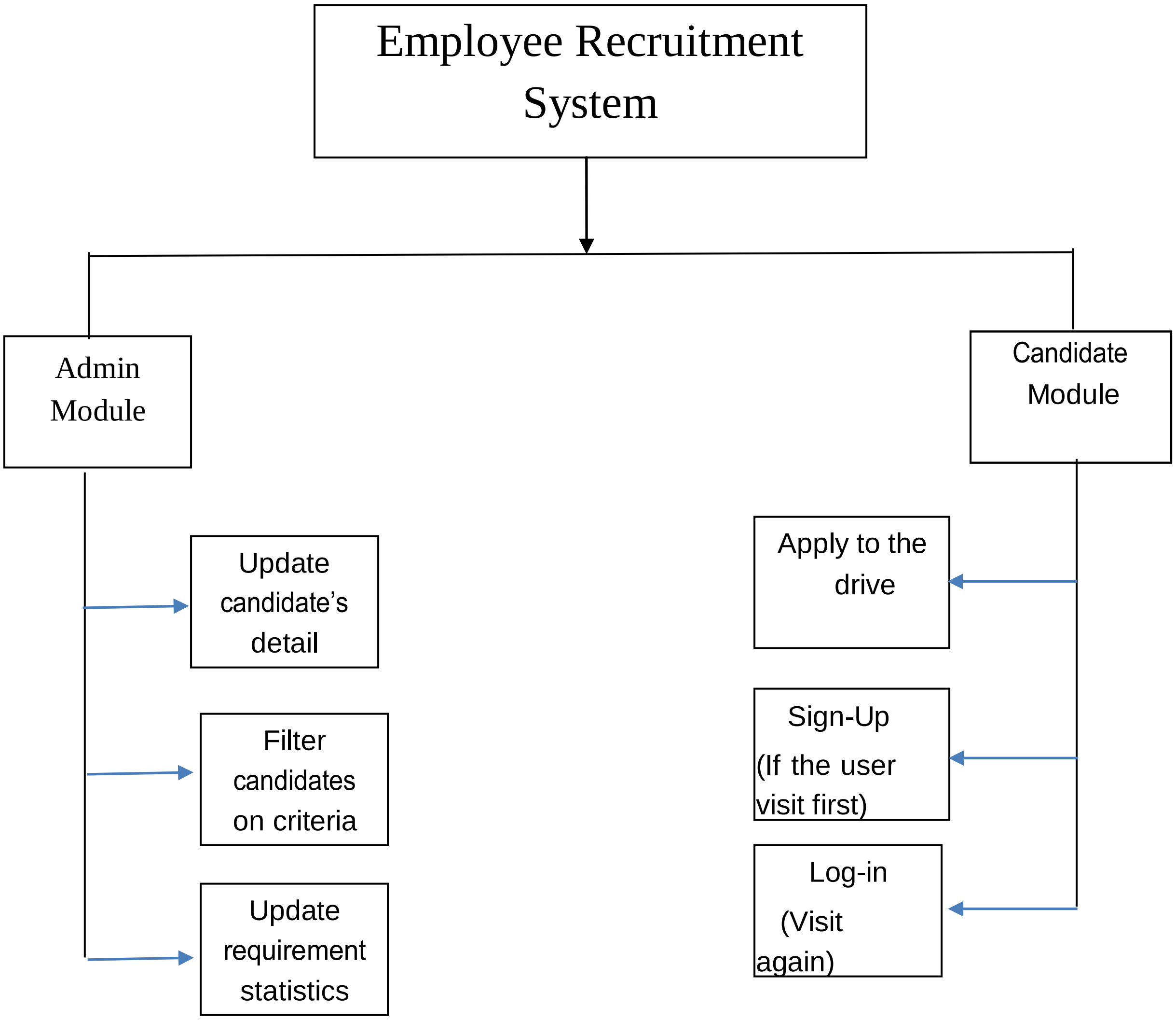
.

Non-functional requirements:

- This application is secure for every kind of its users, because here is facility of session management. If any user logout from any session then nobody will be able to access his profile without knowing his confidential password.
- The database used here is robust, reliable & fast. So users will have to wait for the output very short time.
- This application can be accessed from any type of platform.
- There is no case of redundancy in the database so it will not take extra memory space.

.

Structural Diagram


Employee Recruitment System
Admin Module
Candidate Module
Update candidate's detail
Filter candidates on criteria
Update requirement statistics
Apply to the drive
Sign-Up (If the user visit first)
Log-in (Visit again)

4.1 Structural Diagram

List of Tables

Table: Personal Details (Applicant)

Field	Data Type	Size	Constraints	Description
Username	string	100	Primary Key Auto Increment	It is the unique id of user.
First name	string	255		First name of user
Password	varchar	255		Password for user login
Last name	varchar	255		Last name of applicant
Email	varchar	255		Email Id of the applicant
Address	varchar	255		Applicant's address
Phone no.	int	8		Phone number of the applicant

Table: Admin Details

Field	Data Type	Size	Constraint	Description
Admin id	int	100	Primary Key	It is the unique id of admin.
Password	int	255		Password defined by admin

Conclusions

These are our main recommendations:

- Record all relevant information generated by the project:
  - o Use a notebook,
  - o Keep a diary

- Gather further material from publications or other external resources.

- Organize the material into sections agreed with your supervisor, e.g. Background, and so on.

- Turn this material into written prose to form the project report's main body.

- Where appropriate use
  - o cross-references,
  - o references,
  - o figures and other descriptive devices.

- Produce all required supporting structures according to convention, after completing the main body, and include this material in appendices to avoid disrupting the flow of your narrative.

- For examples to follow, look at o textbooks from reputable publishers, o the way this guide is written.

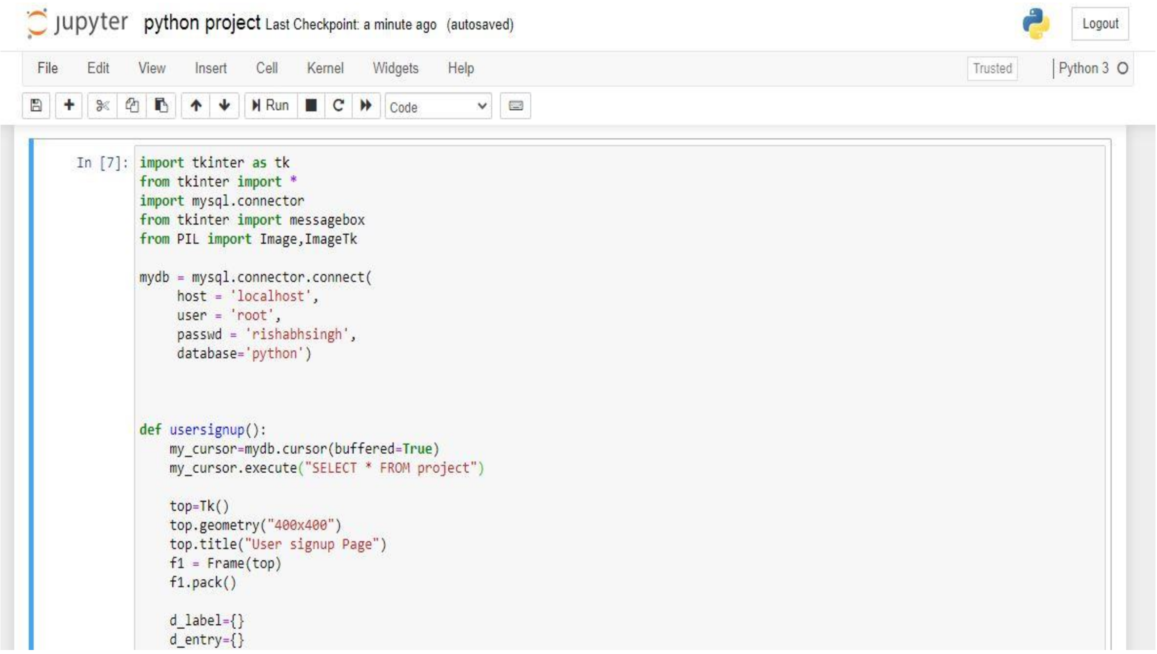
- Discuss an outline of the project report with your supervisor before you begin to write up; this will help you to plan your project. However, we strongly recommend that you write up your work as much as possible as you carry out your project, rather than leaving the writing to last.

If you have any comments, or wish to suggest good sources of advice that you have found, please let your project supervisor know so that we can update these guidelines.

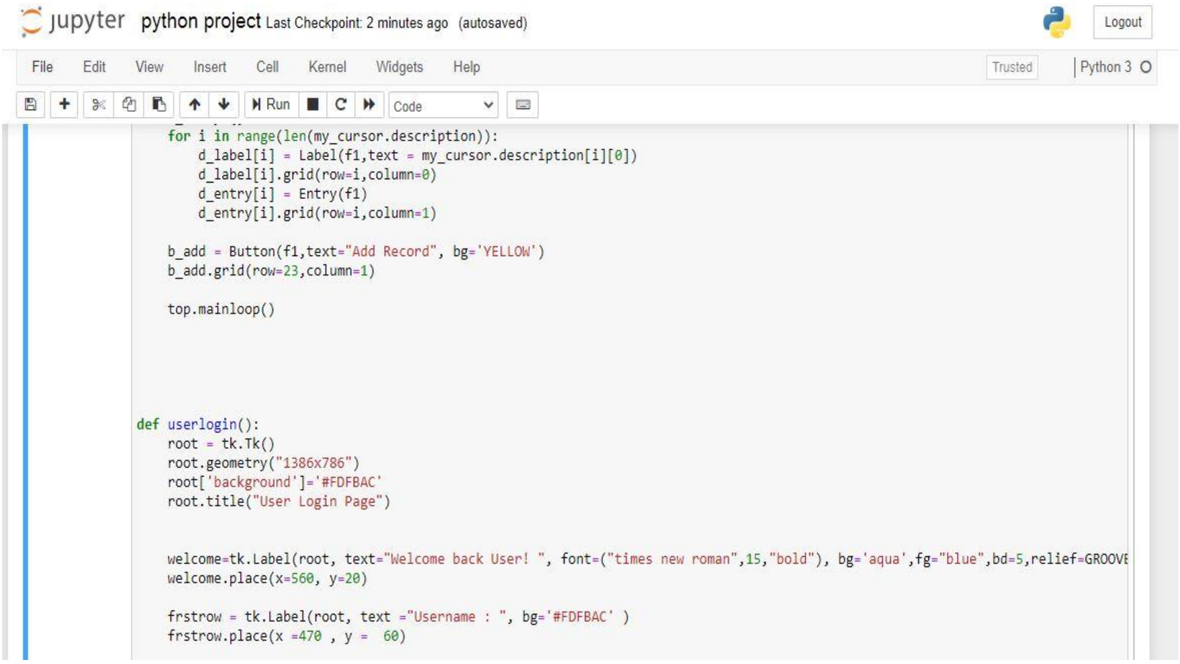
Bibliography

1. http://www.geeksforgeeks.org/
2. http://stackoverflow.com/
3. http://www.google.com/
4. http://www.python-course.eu/
5. http://www.c-sharpcorner.com/
6. http://www.w3schools.com/
7. http://mysql-com.en.softonic.com/

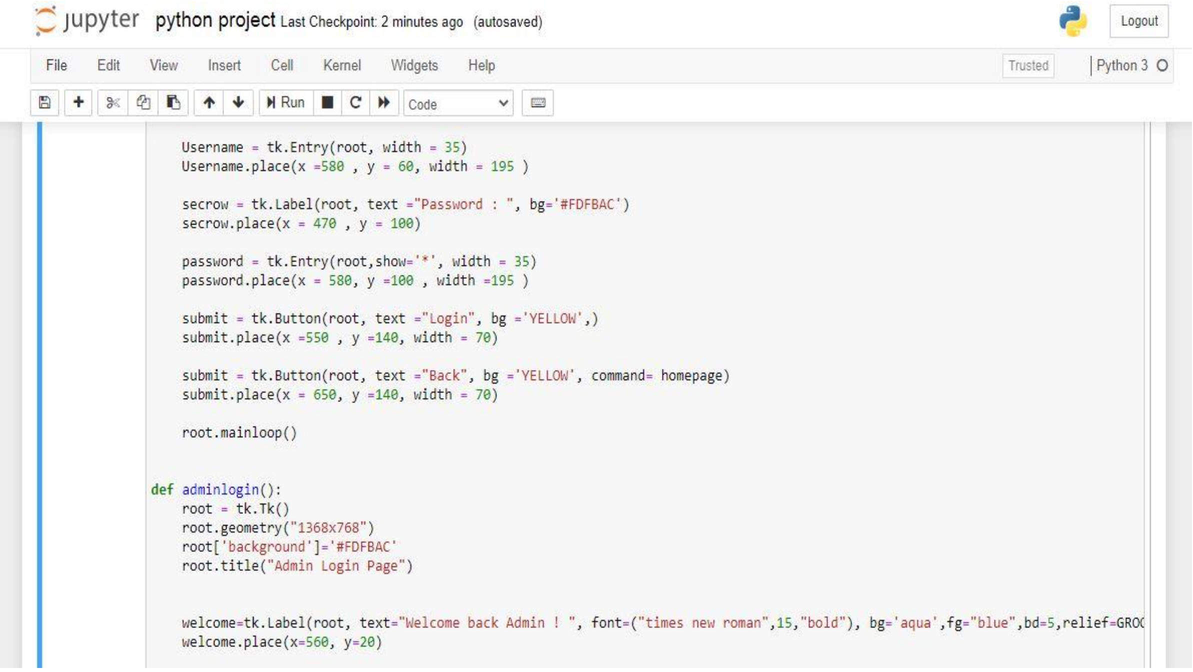
Source Code:



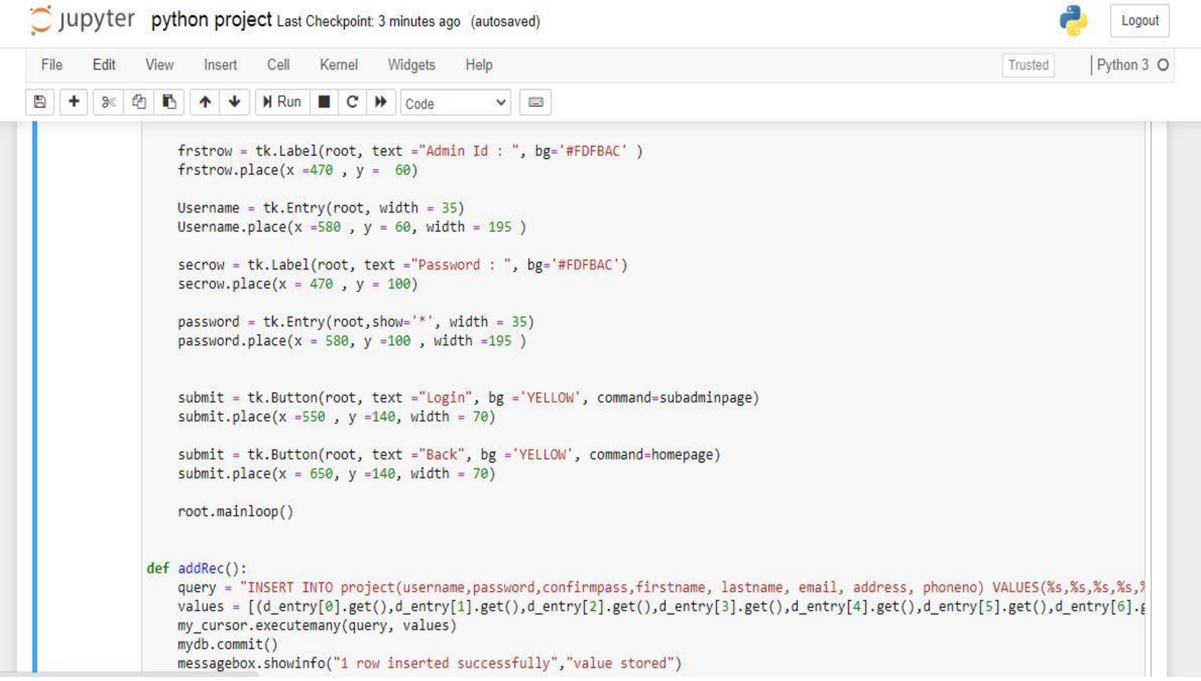
```
import tkinter as tk
from tkinter import *
import mysql.connector
from tkinter import messagebox
from PIL import Image,ImageTk

mydb = mysql.connector.connect(
     host = 'localhost',
     user = 'root',
     passwd = 'rishabhsingh',
     database='python')



def usersignup():
    my_cursor=mydb.cursor(buffered=True)
    my_cursor.execute("SELECT * FROM project")

    top=Tk()
    top.geometry("400x400")
    top.title("User signup Page")
    f1 = Frame(top)
    f1.pack()

    d_label={}
    d_entry={}
```



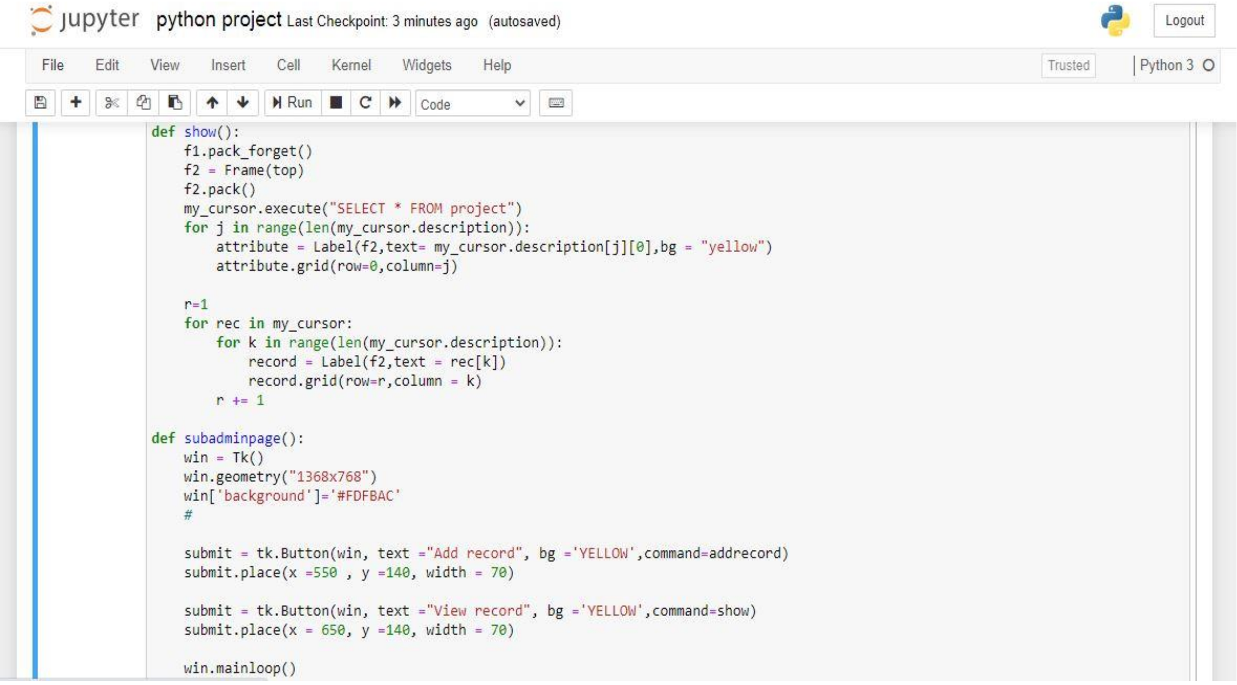
```
    for i in range(len(my_cursor.description)):
        d_label[i] = Label(f1,text = my_cursor.description[i][0])
        d_label[i].grid(row=i,column=0)
        d_entry[i] = Entry(f1)
        d_entry[i].grid(row=i,column=1)

    b_add = Button(f1,text="Add Record", bg='YELLOW')
    b_add.grid(row=23,column=1)

    top.mainloop()




def userlogin():
    root = tk.Tk()
    root.geometry("1386x786")
    root['background']='#FDFBAC'
    root.title("User Login Page")


    welcome=tk.Label(root, text="Welcome back User! ", font=("times new roman",15,"bold"), bg='aqua',fg="blue",bd=5,relief=GROOVE
    welcome.place(x=560, y=20)

    frstrow = tk.Label(root, text ="Username : ", bg='#FDFBAC' )
    frstrow.place(x =470 , y =  60)
```


jupyter python project Last Checkpoint: 2 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

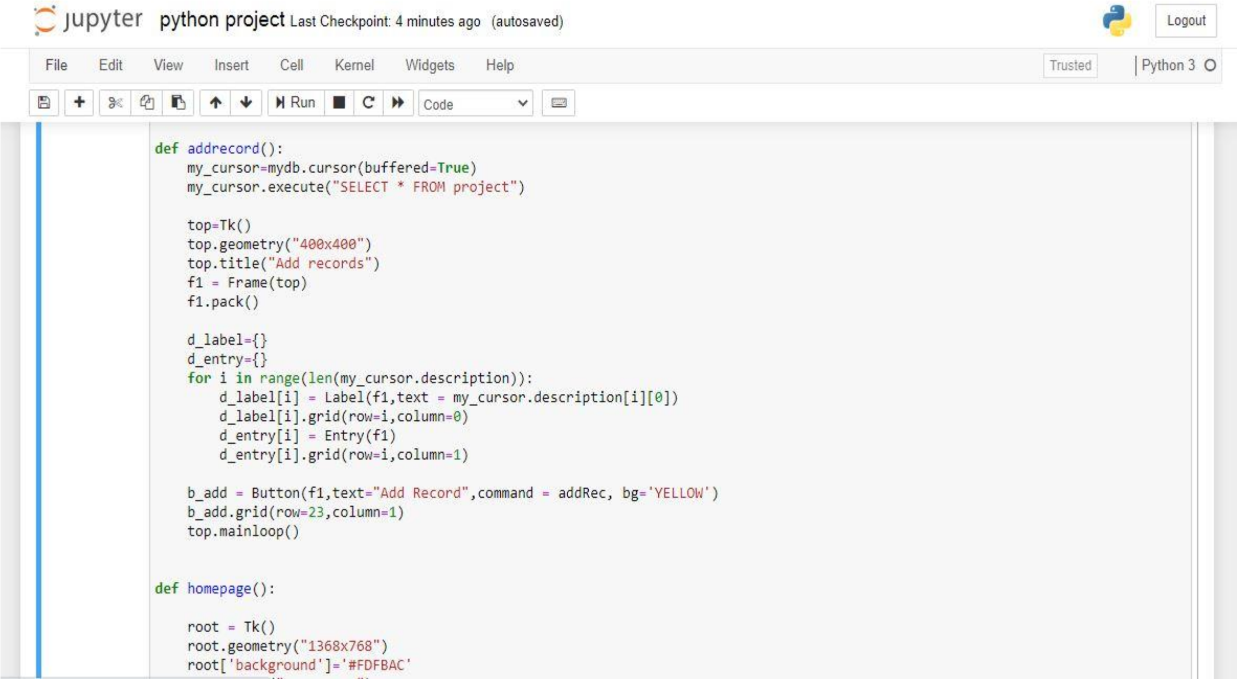
```
    Username = tk.Entry(root, width = 35)
    Username.place(x =580 , y = 60, width = 195 )

    secrow = tk.Label(root, text ="Password : ", bg='#FDFBAC')
    secrow.place(x = 470 , y = 100)

    password = tk.Entry(root,show='*', width = 35)
    password.place(x = 580, y =100 , width =195 )

    submit = tk.Button(root, text ="Login", bg ='YELLOW',)
    submit.place(x =550 , y =140, width = 70)

    submit = tk.Button(root, text ="Back", bg ='YELLOW', command= homepage)
    submit.place(x = 650, y =140, width = 70)

    root.mainloop()


def adminlogin():
    root = tk.Tk()
    root.geometry("1368x768")
    root['background']='#FDFBAC'
    root.title("Admin Login Page")


    welcome=tk.Label(root, text="Welcome back Admin ! ", font=("times new roman",15,"bold"), bg='aqua',fg="blue",bd=5,relief=GROO
    welcome.place(x=560, y=20)
```


jupyter python project Last Checkpoint: 3 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

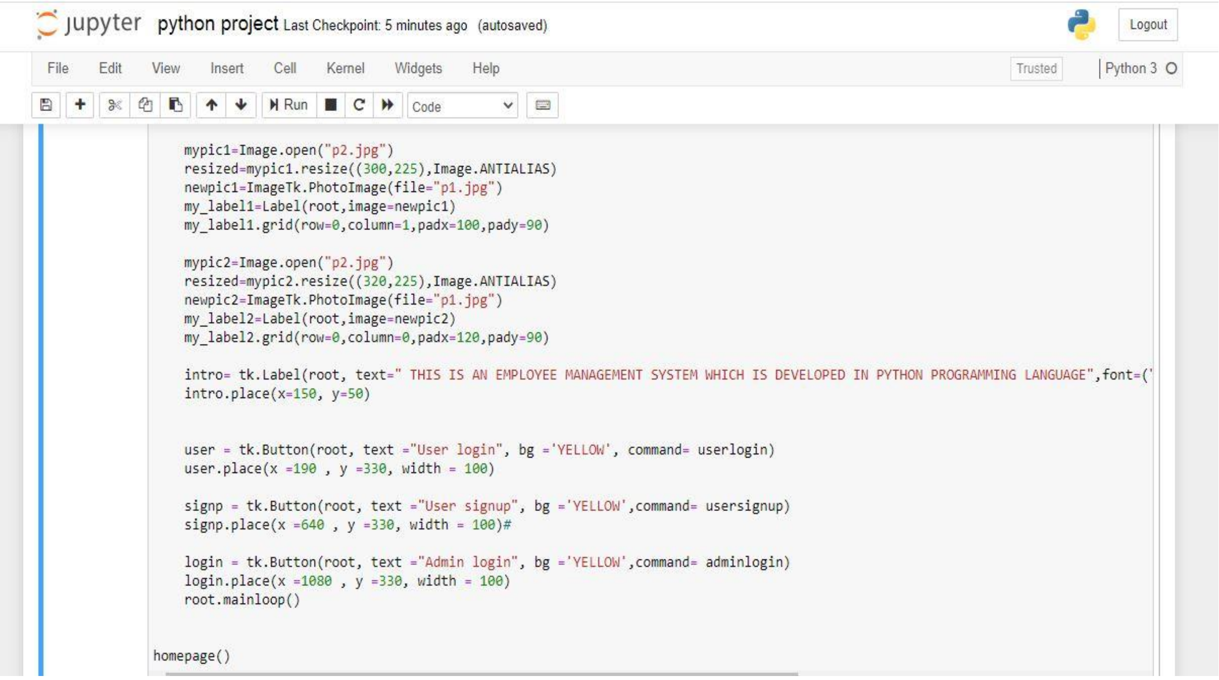
```
    frstrow = tk.Label(root, text ="Admin Id : ", bg='#FDFBAC' )
    frstrow.place(x =470 , y =  60)

    Username = tk.Entry(root, width = 35)
    Username.place(x =580 , y = 60, width = 195 )

    secrow = tk.Label(root, text ="Password : ", bg='#FDFBAC')
    secrow.place(x = 470 , y = 100)

    password = tk.Entry(root,show='*', width = 35)
    password.place(x = 580, y =100 , width =195 )


    submit = tk.Button(root, text ="Login", bg ='YELLOW', command=subadminpage)
    submit.place(x =550 , y =140, width = 70)

    submit = tk.Button(root, text ="Back", bg ='YELLOW', command=homepage)
    submit.place(x = 650, y =140, width = 70)

    root.mainloop()


def addRec():
    query = "INSERT INTO project(username,password,confirmpass,firstname, lastname, email, address, phoneno) VALUES(%s,%s,%s,%s,%
    values = [(d_entry[0].get(),d_entry[1].get(),d_entry[2].get(),d_entry[3].get(),d_entry[4].get(),d_entry[5].get(),d_entry[6].g
    my_cursor.executemany(query, values)
    mydb.commit()
    messagebox.showinfo("1 row inserted successfully","value stored")
```



```
def show():
    f1.pack_forget()
    f2 = Frame(top)
    f2.pack()
    my_cursor.execute("SELECT * FROM project")
    for j in range(len(my_cursor.description)):
        attribute = Label(f2,text= my_cursor.description[j][0],bg = "yellow")
        attribute.grid(row=0,column=j)

    r=1
    for rec in my_cursor:
        for k in range(len(my_cursor.description)):
            record = Label(f2,text = rec[k])
            record.grid(row=r,column = k)
        r += 1

def subadminpage():
    win = Tk()
    win.geometry("1368x768")
    win['background']='#FDFBAC'
    #

    submit = tk.Button(win, text ="Add record", bg ='YELLOW',command=addrecord)
    submit.place(x =550 , y =140, width = 70)

    submit = tk.Button(win, text ="View record", bg ='YELLOW',command=show)
    submit.place(x = 650, y =140, width = 70)

    win.mainloop()
```

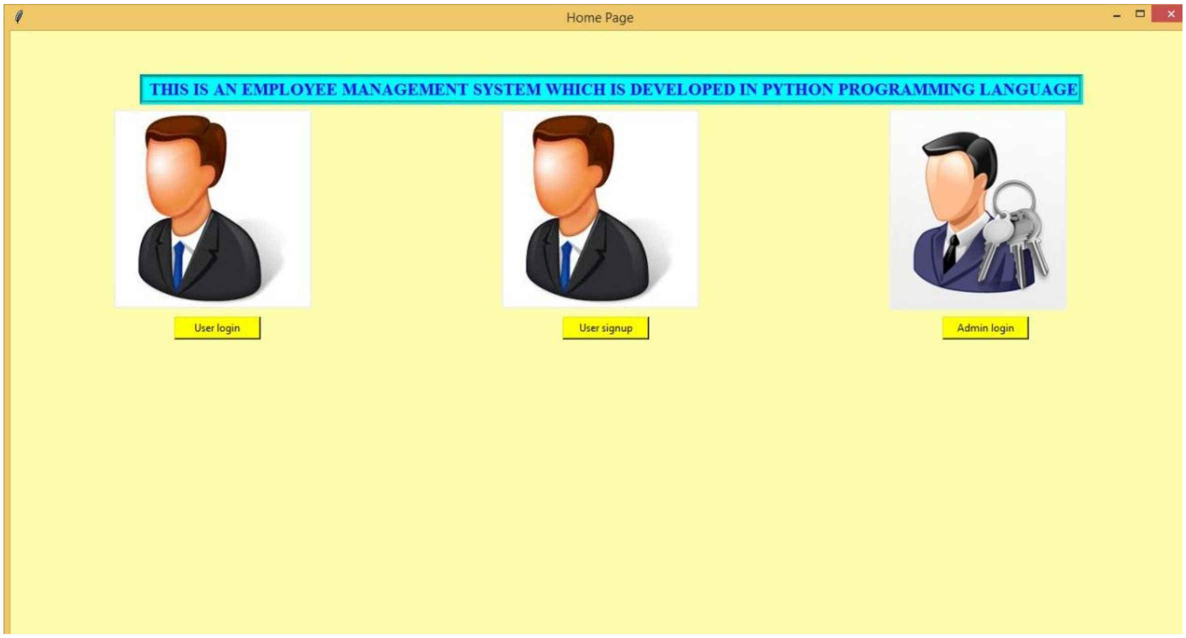


```
def addrecord():
    my_cursor=mydb.cursor(buffered=True)
    my_cursor.execute("SELECT * FROM project")

    top=Tk()
    top.geometry("400x400")
    top.title("Add records")
    f1 = Frame(top)
    f1.pack()

    d_label={}
    d_entry={}
    for i in range(len(my_cursor.description)):
        d_label[i] = Label(f1,text = my_cursor.description[i][0])
        d_label[i].grid(row=i,column=0)
        d_entry[i] = Entry(f1)
        d_entry[i].grid(row=i,column=1)

    b_add = Button(f1,text="Add Record",command = addRec, bg='YELLOW')
    b_add.grid(row=23,column=1)
    top.mainloop()


def homepage():

    root = Tk()
    root.geometry("1368x768")
    root['background']='#FDFBAC'
```



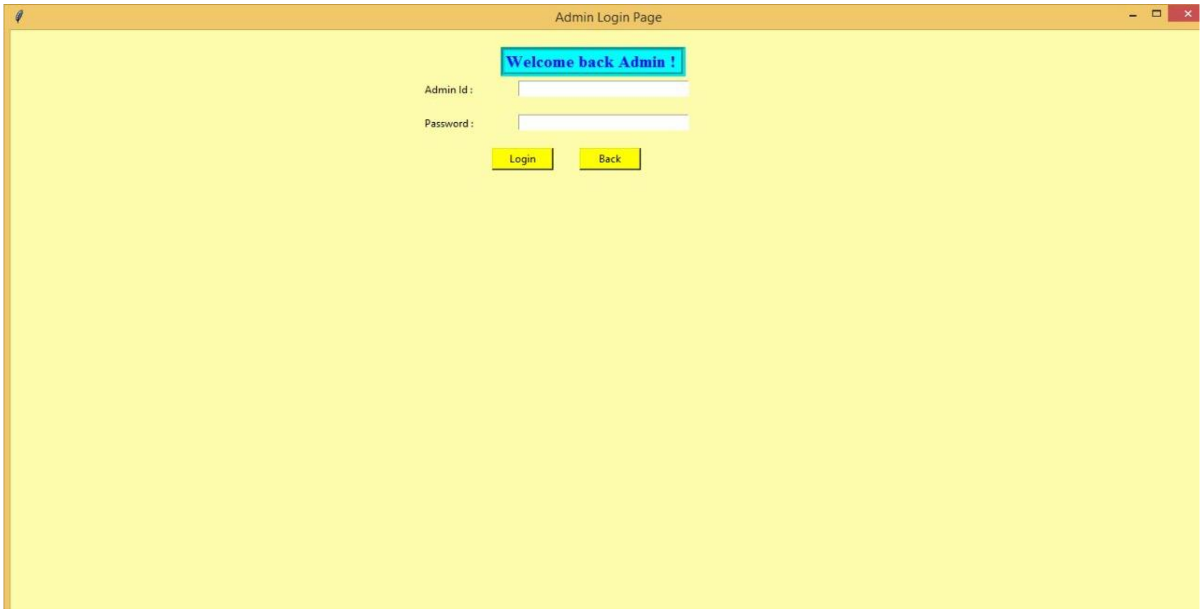
```
    mypic1=Image.open("p2.jpg")
    resized=mypic1.resize((300,225),Image.ANTIALIAS)
    newpic1=ImageTk.PhotoImage(file="p1.jpg")
    my_label1=Label(root,image=newpic1)
    my_label1.grid(row=0,column=1,padx=100,pady=90)

    mypic2=Image.open("p2.jpg")
    resized=mypic2.resize((320,225),Image.ANTIALIAS)
    newpic2=ImageTk.PhotoImage(file="p1.jpg")
    my_label2=Label(root,image=newpic2)
    my_label2.grid(row=0,column=0,padx=120,pady=90)

    intro= tk.Label(root, text=" THIS IS AN EMPLOYEE MANAGEMENT SYSTEM WHICH IS DEVELOPED IN PYTHON PROGRAMMING LANGUAGE",font=('
    intro.place(x=150, y=50)


    user = tk.Button(root, text ="User login", bg ='YELLOW', command= userlogin)
    user.place(x =190 , y =330, width = 100)

    signp = tk.Button(root, text ="User signup", bg ='YELLOW',command= usersignup)
    signp.place(x =640 , y =330, width = 100)#

    login = tk.Button(root, text ="Admin login", bg ='YELLOW',command= adminlogin)
    login.place(x =1080 , y =330, width = 100)
    root.mainloop()


homepage()
```

Output Screen :

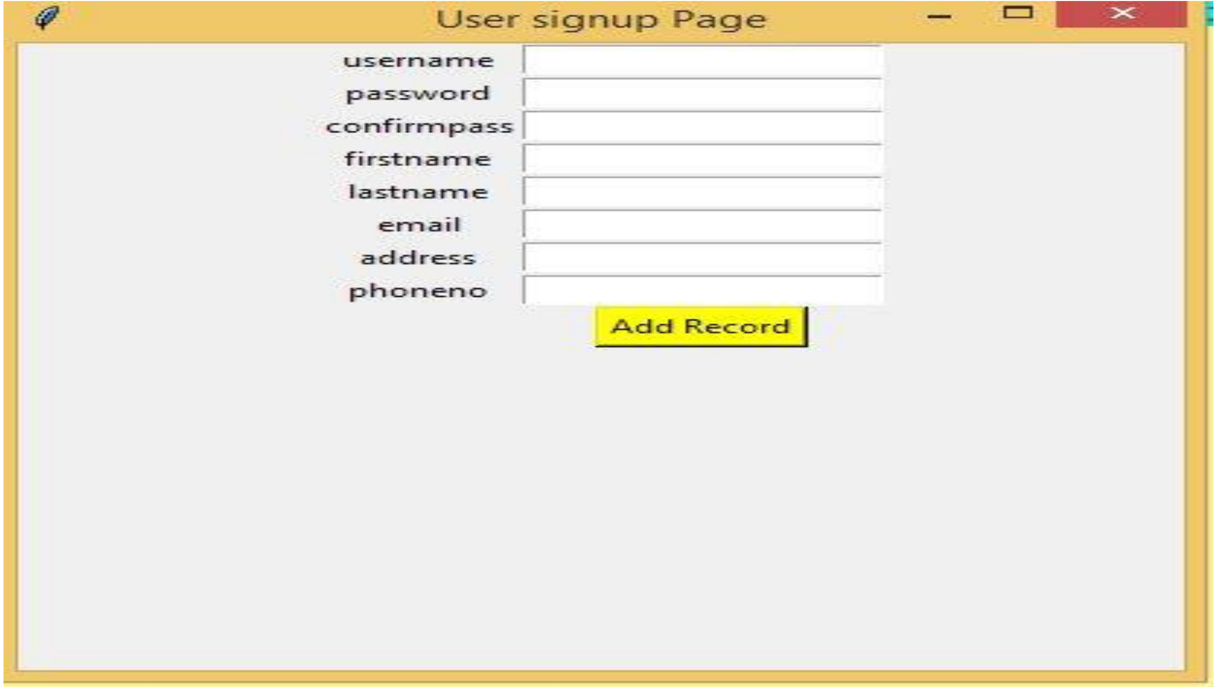


Home Page
THIS IS AN EMPLOYEE MANAGEMENT SYSTEM WHICH IS DEVELOPED IN PYTHON PROGRAMMING LANGUAGE
User login
User signup
Admin login

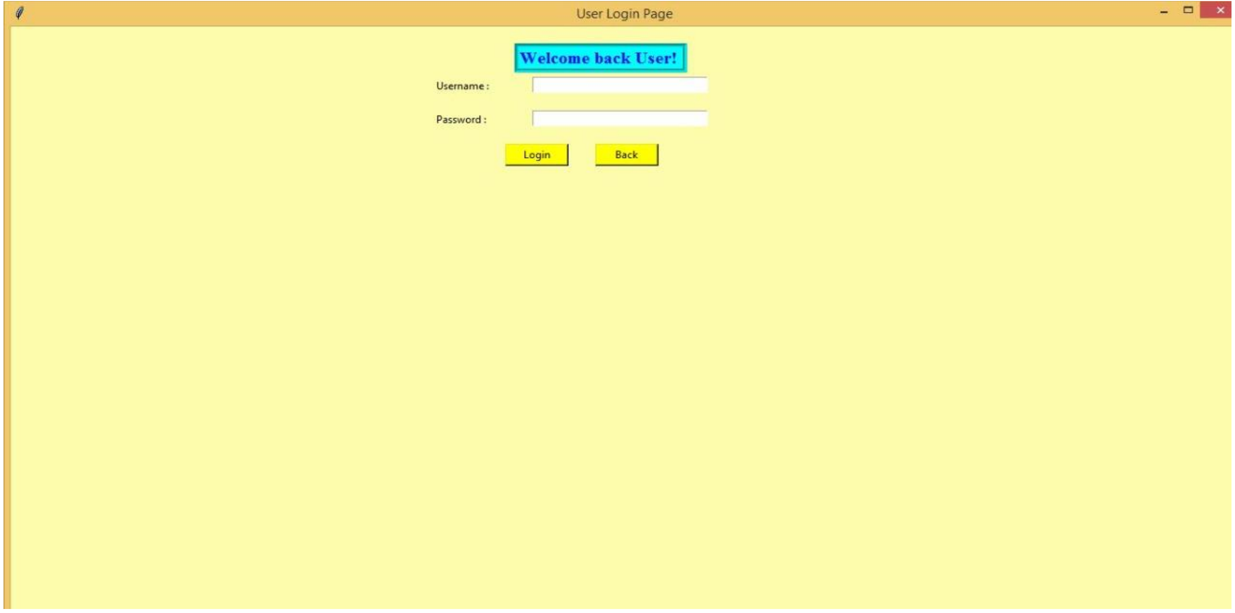
2. GUI for Admin



3.GUI for a New User to Sign-Up



4.GUI for the User who already have a account :



THANK YOU !