

WORK WAGE PROGRAM

- KASHYAP BINU
- KUSHAL B GOWDA
- HAMISH JESUDOS

INDEX

<u><i>S.No</i></u>	<u><i>Topic</i></u>	<u><i>Pg.no</i></u>
<u><i>1.</i></u>	<i>System Hardware and Software Specifications</i>	5
<u><i>2.</i></u>	<i>Project synopsis</i>	6
<u><i>3.</i></u>	<i>Design work</i>	8
<u><i>4.</i></u>	<i>Coding</i>	10
<u><i>5.</i></u>	<i>Output</i>	14
<u><i>6.</i></u>	<i>Bibliography</i>	17

SYSTEM SOFTWARE AND HARDWARE

SPECIFICATIONS SOFTWARE

SOFTWARE

The software used to run the program are :

- Tkinter
- MySQL
- python

HARDWARE

The hardware used to run the project are :

- HP pavilion

PROJECT SYNOPSIS

Aim of “WORK WAGE MANAGEMENT PROGRAM”

Our project “WORK WAGE MANAGEMENT PROGRAM” revolves around calculating the duration of work done by a worker after he/she checks in his/her details and thereby calculating the resultant amount of wage that he/she has earned through that much work .

Introduction:

Millions of workers around the world are paid low wages even though they have worked for long hours. The owners exploit the workers by themselves controlling the amount of wages to be given neglecting the work durations of the workers. Our project “WORK WAGE MANAGEMENT PROGRAM” takes care of this issue since the program itself calculates the work duration of a worker and shows the wage he/ she has earned through their work. This way the owner cannot exploit the workers.

So the basic question is “how does it work?”. Well the answer is simple the program asks the user to input his/her office/work id. Once entered the program shows the details of the user that is preregistered and stored in the database of the system .Once confirmed , the user has to check in by clicking a “check in” button. Once the work is done , the user /worker re enters his/her id to the program and the program again confirms the users details. After confirmation the user must click on the check out button. The program then calculates the time duration of work and then assigns salary to the worker based on the work duration (per hour).

USES OF WORK WAGE PROGRAM:

- *Pays the exact amount of salary that the employee deserves and prevents exploitation by owners.*
- *Stores the employees details and work duration in its database thereby easing work of owner by keeping record of everything.*
- *Keeps a systematic track of everything without any error.*

HARDWARE USED IN WORK WAGE PROGRAM:-

- *Display - shows the user all the information.*
- *Fingerprint sensor*
- *Record printer*
- *Function key buttons*
- *CPU*

Software used in WORK WAGE PROGRAM:-

Typical platforms used in Wage work program include:

- *Python*
- *Tkinter*
- *Microsoft operation system*

DESIGN WORK-

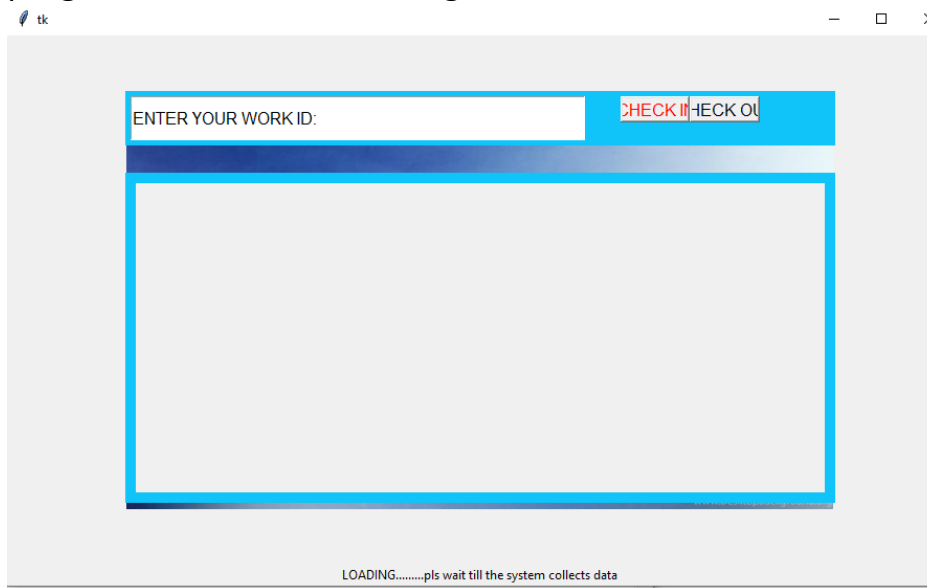
FUNCTION NAME	USE
1. place FUNCTION	Used to place a label or image or any other utility in tkinter.
2. grid FUNCTION	Used to place a tkinter utility according to coordinates assigned.
3. pack FUNCTION	Used to declare position of widgets with relative coordinates
4. entry FUNCTION	Substitute of the input function used In python. Provides space for input from users.
5. button FUNCTION	Makes a GUI button in python Tkinter .
6. execute FUNCTION	Runs SQL statements or stored procedures from a package.
7. commit FUNCTION	Used to permanently save the changes done in the databases.

EXECUTION-

The project has been divided into 2 parts-

1. GUI- using python tkinter-

The page that we have created has been made using python tkinter through use of various widgets and functions like labels, buttons, frame , layout etc. The program GUI looks something like this-



2. DATABASE- using mySQL –

The main calculations and storage utility is provided using mySQL coding. Each data of the employee is made to store in the system and is shown up as soon as the id is entered on the entry provided on the page(made using tkinter).the calculation of work time and the resultant wage is also taken care by this section of coding.

SOURCE CODE-

```
import mysql.connector
from tkinter import *
import tkinter as tk
from PIL import ImageTk, Image
import time
mydb=mysql.connector.connect(host="localhost", user="root", password="1234", database="workers")
"""

#kbg.execute("CREATE DATABASE workers")

#CREATING TABLE
kbg.execute("CREATE TABLE wagedata(id VARCHAR(255), name VARCHAR(255), wage VARCHAR(255),
checkin VARCHAR(255), checkout VARCHAR(255))")
mydb.commit()
"""

kbg=mydb.cursor()
#CODING FOR ADDING NEW WORKER
"""

def check():#ADD NEW WORKER
    main_data=("INSERT INTO wagedata(id, name) VALUES(%s, %s)")
    a=input("ENTER NAME")
    main_data1=(entry.get(), a)
    kbg.execute(main_data, main_data1)
    mydb.commit()"""

"-----"
-----"

#TKINTER WINDOW FRAMES AND ETC
import tkinter as tk
height= 500
width=600
root=tk.Tk()
canvas=tk.Canvas(root, height=height, width=width,bg="blue")
canvas.pack()
background_image=tk.PhotoImage(file="E:\Downloads\BG.png")
background_label=tk.Label(root,image=background_image)
background_label.place(relwidth=1,relheight=1)
frame=tk.Frame(root,bg="#10C3F8",bd=5)
frame.place(relx=0.5,rely=0.1,relwidth=0.75,relheight=0.12,anchor="n")
entry=tk.Entry(frame,font=40)
entry.place(relwidth=0.65,relheight=1)
entry.insert(0,"ENTER YOUR WORK ID:")

"-----"
-----"
```



```

def check_in():#CODING FOR CHECK IN BUTTON
    global wagetime
    update="UPDATE wagedata SET checkin= %s WHERE id= %s"#UPDATES THE CHECK IN TIME
    id_3=(time.asctime(), entry.get() )
    kbg.execute(update, id_3)
    id_1="SELECT * FROM wagedata WHERE id=%s"
    id_2=(entry.get(),)
    kbg.execute(id_1, id_2)
    result=kbg.fetchall()
    for i in result:
        label=tk.Label(lower_frame, text="ID:"+str(i[0]), height=2, width=125, font=("Times", 12, "bold italic"),
        anchor='center', borderwidth=2,relief="solid")
        label.grid(row=2, column=0)

        label2=tk.Label(lower_frame, bg="#10C3F8")
        label2.grid(row=3, column=0)

        label3=tk.Label(lower_frame, text="NAME:"+i[1], height=2, width=125, font=("Times", 12, "bold
        italic"), anchor='center', borderwidth=2, relief="solid")
        label3.grid(row=4, column=0)

        label2=tk.Label(lower_frame, bg="#10C3F8")
        label2.grid(row=5, column=0)

        label3=tk.Label(lower_frame, text="CHECK IN TIME:"+i[3], height=2, width=125, font=("Times", 12,
        "bold italic"), anchor='center', borderwidth=2, relief="solid")
        label3.grid(row=6, column=0)
    mydb.commit()
    wagetime=time.time() #NOTES DOWN THE DECIMAL TIME VALUE
    "#-----"
    "-----"

def check_out():#CODING FOR CHECK OUT BUTTON
    global wagetime2
    update="UPDATE wagedata SET checkout= %s WHERE id= %s"#UPDATES THE TIMNE AT CHECKOUT
    id_4=(time.asctime(), entry.get() )
    kbg.execute(update, id_4)
    id_5="SELECT * FROM wagedata WHERE id=%s"
    id_6=(entry.get(),)
    kbg.execute(id_5, id_6)
    result=kbg.fetchall()
    for j in result:
        label=tk.Label(lower_frame, text="ID:"+str(j[0]), height=2, width=125, font=("Times", 12, "bold italic"),
        anchor='center', borderwidth=2,relief="solid")
        label.grid(row=2, column=0)

        label2=tk.Label(lower_frame, bg="#10C3F8")
        label2.grid(row=3, column=0)

        label3=tk.Label(lower_frame, text="NAME:"+j[1], height=2, width=125, font=("Times", 12, "bold

```

```

italic"), anchor='center', borderwidth=2, relief="solid")
    label3.grid(row=4, column=0)

    label4=tk.Label(lower_frame, bg="#10C3F8")
    label4.grid(row=5, column=0)

    label5=tk.Label(lower_frame, text="CHECK IN TIME:"+j[3], height=2, width=125, font=("Times", 12,
"bold italic"), anchor='center', borderwidth=2, relief="solid")
    label5.grid(row=6, column=0)

    label6=tk.Label(lower_frame, bg="#10C3F8")
    label6.grid(row=7, column=0)

    label7=tk.Label(lower_frame, text="CHECK OUT TIME:"+j[4], height=2, width=125, font=("Times", 12,
"bold italic"), anchor='center', borderwidth=2, relief="solid")
    label7.grid(row=8, column=0)
    button=tk.Button(frame,text="CHECK IN",fg="red",font=35,command=check_in, state=DISABLED)
    button.place(relx=0.7,relwidth=0.1,relheight=0.6)
    wagetime2=time.time()
    mydb.commit()

"#####
-----"

def wage_1():#CODING FOR CALCULATING AND DISPLAYING THE MONEY EARNED
    timer=int(wagetime2-wagetime)
    wages=timer*1000 #UPDATE THE WAGE PER HOUR
    update="UPDATE wagedata SET wage= %s WHERE id= %s"
    id_7=(wages, entry.get() )
    kbg.execute(update, id_7)
    id_8="SELECT * FROM wagedata WHERE id=%s"
    id_9=(entry.get(),)
    kbg.execute(id_8, id_9)
    result=kgb.fetchall()
    for k in result:
        label8=tk.Label(lower_frame, bg="#10C3F8")
        label8.grid(row=9, column=0)

        label9=tk.Label(lower_frame, text="TODAYS EARNINGS:"+k[2], height=2, width=125, font=("Times", 12,
"bold italic"), anchor='center', borderwidth=2, relief="solid")
        label9.grid(row=10, column=0)
        mydb.commit()
def clear():#CODING FOR CLEAR BUTTON
    button=tk.Button(frame,text="CHECK IN",fg="red",font=35,command=check_in, state=NORMAL)
    button.place(relx=0.7,relwidth=0.1,relheight=0.6)
    for widget in lower_frame.winfo_children():
        widget.destroy()

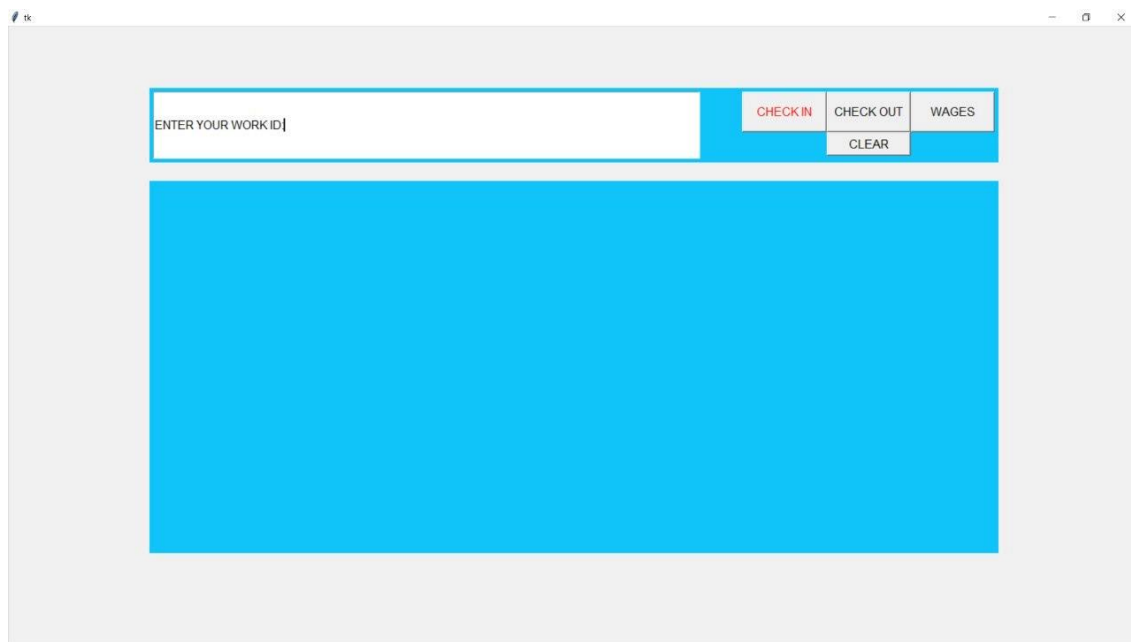
```

#ALL THE BUTTONS LIST

```
"#-----  
-----"  
  
button=tk.Button(frame,text="CHECK IN",fg="red",font=40,command=check_in)  
button.place(relx=0.7,relwidth=0.1,relheight=0.6)  
"#-----  
-----"  
  
button2=tk.Button(frame,text="CHECK OUT",font=40, command=check_out)  
button2.place(relx=0.8,relwidth=0.1,relheight=0.6)  
"#-----  
-----"  
  
button3=tk.Button(frame,text="WAGES",font=40, command=wage_1)  
button3.place(relx=0.9,relwidth=0.1,relheight=0.6)  
"#-----  
-----"  
  
button4=tk.Button(frame,text="CLEAR",font=40, command=clear, height=1,)  
button4.place(relx=0.8, rely=0.6, relwidth=0.1)  
"#-----  
-----"  
  
lower_frame=tk.Frame(root,bg="#10C3F8",bd=10)  
lower_frame.place(relx=0.5,rely=0.25,relwidth=0.75,relheight=0.6,anchor="n")  
  
root.mainloop()
```

OUTPUT:

1. Interface asking user to enter id.



A screenshot of a Tkinter window titled 'tk'. The window has a light gray background. At the top, there is a white input field with the placeholder text 'ENTER YOUR WORK ID'. To the right of the input field are four buttons: 'CHECK IN' (with a red border), 'CHECK OUT', 'WAGES', and 'CLEAR'. Below the input field and buttons is a large blue rectangular area.

2. Enters id and clicks on check in- program shows check in time.



A screenshot of the same Tkinter window after the 'CHECK IN' button has been clicked. The input field now contains the text '1000'. Below the input field and buttons, three horizontal white bars with black borders display the following information: 'ID:1000', 'NAME:KUSHAL', and 'CHECK IN TIME:Mon Jan 3 17:45:16 2022'. The large blue rectangular area remains at the bottom.

3. Clicks on check out after work. Program displays check out time.

1000

CHECK IN	CHECK OUT	WAGES
CLEAR		

ID:1000

NAME:KUSHAL

CHECK IN TIME:Mon Jan 3 17:45:16 2022

CHECK OUT TIME:Mon Jan 3 17:45:38 2022

4. Wages button results in an output showing the employees current earning .

1000

CHECK IN	CHECK OUT	WAGES
CLEAR		

ID:1000

NAME:KUSHAL

CHECK IN TIME:Mon Jan 3 17:45:16 2022

CHECK OUT TIME:Mon Jan 3 17:45:38 2022

TODAYS EARNINGS:21000

BIBLIOGRAPHY:

SPECIAL THANKS TO-

1. CODEMY YT – *helped in learning basics of tkinter and SQL.*

2. KEITH GALLI – *tkinter GUI basics*