**System Implementations**

**Recommended System Requirements**

Processors: Intel® Core™ i3 processor 4300M at 2.60 GHz.

Disk space: 4 to 8 GB.

Operating systems: Windows® 10, MACOS, and UBUNTU.

Python Versions: 3.X.X or Higher.

**Minimum System Requirements**

Processors: Intel Atom® processor or Intel® Core™ i3 processor.

Disk space: 1 GB.

Operating systems: Windows 7 or later, MACOS, and UBUNTU.

Python Versions: 2.7.X, 3.9.X.

**ACKNOWLEDGEMENT**TTT

First and foremost, praises and thanks to the God, the Almighty, for His showers of blessings throughout my research work to complete the research successfully.

We would like to express my deep and sincere gratitude to my subject teacher, Mr. Amit Udiwal, for giving me the opportunity to do research and providing invaluable guidance throughout this research. His dynamism, vision, sincerity and motivation have deeply inspired me. He has taught me the methodology to carry out the research and to present the research works as clearly as and honour to work and study under his guidance. We are very much thankful to our Sr. Renjana for giving valuable time and moral support to develop this software. We would like to take opportunity to extend my sincere thanks and gratitude to our parents for being a source of inspiration and providing time and freedom to develop this software project. We also feel indebted to my friends for the valuable suggestions during the project work.

Mahi Dawre

[Roll No.

Class XII

**CERTIFICATE**

This is to certify that the project on ‘Employee Management System’ is a work done by Mahi Dawre fulfilment of CBSE’S AISSCE EXAMINATION 2020¢and has been carried out under my direct supervision and guidance. This report or a similar report on the topic has not been submitted for any other examination and does not form any other examination and does not form any other course undergone by the candidate.

Name:

Mahi Dawre [Roll No.

………………….

Signature of Teacher / Guide

Name: Mr. Amit Udiwal

Designation:

………………. ….………………

**REFERENCE**

The order to work on this project on ‘Sales And Inventory Management System’ the following books & literature are referred by me during the various phrases of department of the project.

• http://www.python.org/.

• http://www.itsourcecode.org/.

• http://www.wikipedia.org/.

• Informatics Practices for Class XII

- By Sumita Arora

• Together with informatics practices.

Other than the above mentioned books, the suggestions and supervision of my teacher and my class experience also helped me to develop this software project.

**Introduction**

A **Employee Management System Project in Python** is created in Python Programming Language using PyCharm Community IDE, This Simple Project was developed using console based and connected into MySQL Database as the system’s Back-end.

The idea is that we use different functions to perform different changes in our Employee Record, for example, the Add Employee function will insert a new row in our Employee table, and we will also create a Remove Employee Function that will delete the record of any existing employee in our Employee table.

**Objective and**

**Scope of The Project**

**Employee Management System** aids in the automation of manual processes, saving both time and money. This system safeguards the professional and personal information of employees and the firm. HR and business managers are relieved of their burdens and pressures thanks to the personnel management system.

***Functions:***

* Add Employee
* Remove Employee
* Update Employee
* Display Employees

**Employee Management System**

from tkinter import \*

from tkinter import ttk

from tkinter import messagebox

from db import Database

db = Database("Employee.db")

root = Tk()

root.title("Employee Management System")

root.geometry("1920x1080+0+0")

root.config(bg="#2c3e50")

root.state("zoomed")

name = StringVar()

age = StringVar()

doj = StringVar()

gender = StringVar()

email = StringVar()

contact = StringVar()

# Entries Frame

entries\_frame = Frame(root, bg="#535c68")

entries\_frame.pack(side=TOP, fill=X)

title = Label(entries\_frame, text="Employee Management System", font=("Calibri", 18, "bold"), bg="#535c68", fg="white")

title.grid(row=0, columnspan=2, padx=10, pady=20, sticky="w")

lblName = Label(entries\_frame, text="Name", font=("Calibri", 16), bg="#535c68", fg="white")

lblName.grid(row=1, column=0, padx=10, pady=10, sticky="w")

txtName = Entry(entries\_frame, textvariable=name, font=("Calibri", 16), width=30)

txtName.grid(row=1, column=1, padx=10, pady=10, sticky="w")

lblAge = Label(entries\_frame, text="Age", font=("Calibri", 16), bg="#535c68", fg="white")

lblAge.grid(row=1, column=2, padx=10, pady=10, sticky="w")

txtAge = Entry(entries\_frame, textvariable=age, font=("Calibri", 16), width=30)

txtAge.grid(row=1, column=3, padx=10, pady=10, sticky="w")

lbldoj = Label(entries\_frame, text="D.O.J", font=("Calibri", 16), bg="#535c68", fg="white")

lbldoj.grid(row=2, column=0, padx=10, pady=10, sticky="w")

txtDoj = Entry(entries\_frame, textvariable=doj, font=("Calibri", 16), width=30)

txtDoj.grid(row=2, column=1, padx=10, pady=10, sticky="w")

lblEmail = Label(entries\_frame, text="Email", font=("Calibri", 16), bg="#535c68", fg="white")

lblEmail.grid(row=2, column=2, padx=10, pady=10, sticky="w")

txtEmail = Entry(entries\_frame, textvariable=email, font=("Calibri", 16), width=30)

txtEmail.grid(row=2, column=3, padx=10, pady=10, sticky="w")

lblGender = Label(entries\_frame, text="Gender", font=("Calibri", 16), bg="#535c68", fg="white")

lblGender.grid(row=3, column=0, padx=10, pady=10, sticky="w")

comboGender = ttk.Combobox(entries\_frame, font=("Calibri", 16), width=28, textvariable=gender, state="readonly")

comboGender['values'] = ("Male", "Female")

comboGender.grid(row=3, column=1, padx=10, sticky="w")

lblContact = Label(entries\_frame, text="Contact No", font=("Calibri", 16), bg="#535c68", fg="white")

lblContact.grid(row=3, column=2, padx=10, pady=10, sticky="w")

txtContact = Entry(entries\_frame, textvariable=contact, font=("Calibri", 16), width=30)

txtContact.grid(row=3, column=3, padx=10, sticky="w")

lblAddress = Label(entries\_frame, text="Address", font=("Calibri", 16), bg="#535c68", fg="white")

lblAddress.grid(row=4, column=0, padx=10, pady=10, sticky="w")

txtAddress = Text(entries\_frame, width=85, height=5, font=("Calibri", 16))

txtAddress.grid(row=5, column=0, columnspan=4, padx=10, sticky="w")

def getData(event):

selected\_row = tv.focus()

data = tv.item(selected\_row)

global row

row = data["values"]

#print(row)

name.set(row[1])

age.set(row[2])

doj.set(row[3])

email.set(row[4])

gender.set(row[5])

contact.set(row[6])

txtAddress.delete(1.0, END)

txtAddress.insert(END, row[7])

def dispalyAll():

tv.delete(\*tv.get\_children())

for row in db.fetch():

tv.insert("", END, values=row)

def add\_employee():

if txtName.get() == "" or txtAge.get() == "" or txtDoj.get() == "" or txtEmail.get() == "" or comboGender.get() == "" or txtContact.get() == "" or txtAddress.get(

1.0, END) == "":

messagebox.showerror("Erorr in Input", "Please Fill All the Details")

return

db.insert(txtName.get(),txtAge.get(), txtDoj.get() , txtEmail.get() ,comboGender.get(), txtContact.get(), txtAddress.get(

1.0, END))

messagebox.showinfo("Success", "Record Inserted")

clearAll()

dispalyAll()

def update\_employee():

if txtName.get() == "" or txtAge.get() == "" or txtDoj.get() == "" or txtEmail.get() == "" or comboGender.get() == "" or txtContact.get() == "" or txtAddress.get(

1.0, END) == "":

messagebox.showerror("Erorr in Input", "Please Fill All the Details")

return

db.update(row[0],txtName.get(), txtAge.get(), txtDoj.get(), txtEmail.get(), comboGender.get(), txtContact.get(),

txtAddress.get(

1.0, END))

messagebox.showinfo("Success", "Record Update")

clearAll()

dispalyAll()

def delete\_employee():

db.remove(row[0])

clearAll()

dispalyAll()

def clearAll():

name.set("")

age.set("")

doj.set("")

gender.set("")

email.set("")

contact.set("")

txtAddress.delete(1.0, END)

btn\_frame = Frame(entries\_frame, bg="#535c68")

btn\_frame.grid(row=6, column=0, columnspan=4, padx=10, pady=10, sticky="w")

btnAdd = Button(btn\_frame, command=add\_employee, text="Add Details", width=15, font=("Calibri", 16, "bold"), fg="white",

bg="#16a085", bd=0).grid(row=0, column=0)

btnEdit = Button(btn\_frame, command=update\_employee, text="Update Details", width=15, font=("Calibri", 16, "bold"),

fg="white", bg="#2980b9",

bd=0).grid(row=0, column=1, padx=10)

btnDelete = Button(btn\_frame, command=delete\_employee, text="Delete Details", width=15, font=("Calibri", 16, "bold"),

fg="white", bg="#c0392b",

bd=0).grid(row=0, column=2, padx=10)

btnClear = Button(btn\_frame, command=clearAll, text="Clear Details", width=15, font=("Calibri", 16, "bold"), fg="white",

bg="#f39c12",

bd=0).grid(row=0, column=3, padx=10)

# Table Frame

tree\_frame = Frame(root, bg="#ecf0f1")

tree\_frame.place(x=0, y=480, width=1980, height=520)

style = ttk.Style()

style.configure("mystyle.Treeview", font=('Calibri', 18),

rowheight=50) # Modify the font of the body

style.configure("mystyle.Treeview.Heading", font=('Calibri', 18)) # Modify the font of the headings

tv = ttk.Treeview(tree\_frame, columns=(1, 2, 3, 4, 5, 6, 7, 8), style="mystyle.Treeview")

tv.heading("1", text="ID")

tv.column("1", width=5)

tv.heading("2", text="Name")

tv.heading("3", text="Age")

tv.column("3", width=5)

tv.heading("4", text="D.O.B")

tv.column("4", width=10)

tv.heading("5", text="Email")

tv.heading("6", text="Gender")

tv.column("6", width=10)

tv.heading("7", text="Contact")

tv.heading("8", text="Address")

tv['show'] = 'headings'

tv.bind("<ButtonRelease-1>", getData)

tv.pack(fill=X)

dispalyAll()

root.mainloop()