

PROFESSIONAL TRAINING REPORT

at

Sathyabama Institute of Science and Technology

(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award
of Bachelor of Engineering Degree in Information Technology

By: V.SRIRANGAPRASANTH

Reg no:39120121



DEPARTMENT |

TECHNOLOGY

SCHOOL OF COMPUTING

**SATHYABAMA INSTITUTE OF SCIENCE AND
TECHNOLOGY**

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI – 600119, TAMILNADU

NOVEMBER2021

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI
CHENNAI– 600119
www.sathyabama.ac.in

DERARTEMENT OF INFORMATION TEHNOLOGY

BONAFIEDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of
V.SRIRANGAPRASANTH (Reg. No: 39120121) who carried out the project entitled
“Employee On Boarding Tool” under my supervision from June 2021 to November
2021.



Internal Guide

Dr.P.Ajitha

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examine

DECLARATION

I, **V.srirangaprasanth** hereby declare that the project report entitled **Employee On Boarding Tool** done by me under the guidance of

DR.P.Ajitha is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Information Technology

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D, Dean**, School of Computing, **Dr.P.Ajitha** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.P.Ajitha** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE



CERTIFICATE OF COMPLETION

CSW-I-23-081121 is given to

SRIRANGAPRASANTH V

For completing the Cyber Security training that includes basics of Cyber Security, Pentesting, Networking, and WebApp.

Aaron V Sandeen

AARON SANDEEN
CEO - CSW

N. Vengatesh

VENGATESH NAGARAJAN
MENTOR



ABSTRACT

In this tutorial, you are going to learn how to create GUI apps in Python. You'll also learn about all the elements needed to develop GUI apps in Python.

Before you begin, you should be familiar with Python to learn Tkinter. If you're new to Python, check out Data Camp's [Introduction to Python](#) course.

TABLE OF CONTENTS		
CHAPTER	TITLE	PAGE NO
	ABSTRACT	5
1	INTRODUCTION	6
2	SYSTEM ANALYSIS	7
	2.1 EXISTING SYSTEM	7
	2.2 PROPOSED SYSTEM	7
3	SYSTEM SPECIFICATION	7
	3.1 HARDWARE SPECIFICATION	7
	3.2 SOFTWARE SPECIFICATION	7
4	SYSTEM IMPLEMENTATION	8
	4.1 SYSTEM ARCHITECTURE	9
	4.2 MODULES	10
5	SYSTEM DESIGN	11
	5.1 DATA FLOW DIAGRAM	11
	5.1.1 LEVEL 0	11
	5.1.2 LEVEL 1	12

	5.1.3 LEVEL 2	13
	5.1.4 LEVEL 3	13
6	CONCLUSION	16
	REFERENCES	17
	APPENDIX A: SNAP SHOTS	11,12,13
	APPENDIX B: SOURCE CODE	14,15,16

CHAPTER 1

INTRODUCTION

- Python offers multiple options for developing GUI (Graphical User Interface)
- Out of all the GUI methods, **Tkinter** is the most commonly used method.
- It is a standard Python interface to the Tk GUI toolkit shipped with Python.
- Python with Tkinter is the fastest and easiest way to create the GUI applications.
- Creating a GUI using Tkinter is an easy task.
- **PostgreSQL** is an enterprise-class open source database management system. It supports both SQL and JSON for relational and non-relational queries for extensibility and SQL compliance. PostgreSQL supports advanced data types and performance optimization features, which are only available in expensive commercial databases, like Oracle and SQL Server. It is also known as Postgres.
- Py GUI is a graphical application cross-platform framework for Unix, Macintosh, and Windows. Compared to some other GUI frameworks, PyGUI is by far the simplest and lightweight of them all, as the API is purely in sync with Python. PyGUI inserts very less code between the GUI platform and Python application; hence, the display of the application usually displays the natural GUI of the platform
- TKINTER commonly comes bundled with Python, using Tk and is Python's standard GUI framework. It is famous for its simplicity and graphical user interface. It is open-source and available under the Python License.

CHAPTER 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

Employee On Boarding tool, used to On Boarding Employee in organization. Tool gets information like First Name, Last Name, Mobile Number, Email ID for employee.

2.2 PROPOSED SYSTEM

Validation implemented for Name, Date of Birth, Mobile Number and Email ID. Stores employee details in secured way using PostgreSQL. Created Db tables to store employee details.

CHAPTER 3

SYSTEM SPECIFICATION

3.1 HARDWARE SPECIFICATION

Processor : Processor Intel CORE i3 and above

Internet Connection : Existing Telephone Lines ,Data card , Fiber net

RAM : 4GB

3.2 SOFTWARE SPECIFICATION

Operating System : Windows

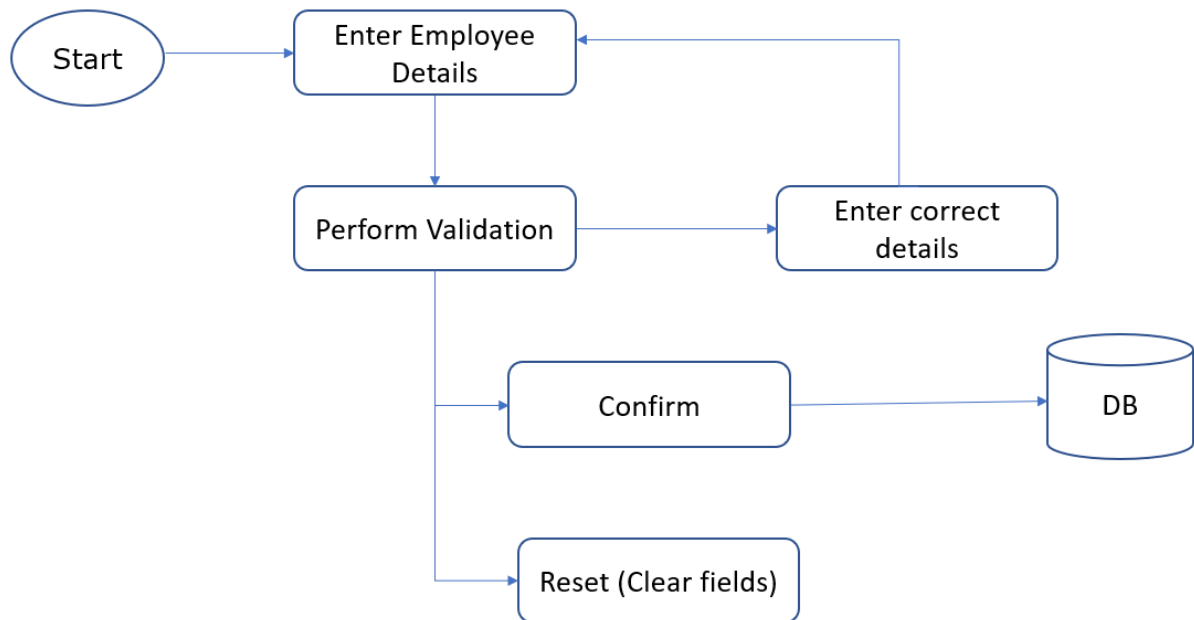
Language : python coding

GUI : Tkinter

CHAPTER 4

SYSTEM IMPLEMENTATION

4.1 SYSTEM ARCHITECTURE



Shows the System Implement diagram of the project define flow charts for classifying the projects includes Start, Employee details, perform validation, confirm, reset (clear fields), Enter correct details and DB includes the diagram. This diagram based on company employee details.

4.2 MODULES

- Tool implemented with 3 modules
 - 1.GUI (**TKINTER**)
 - 2.Implemantion code (**Python**)
 - 3.Database (**PostgreSQL**)

1. GUI (Tkinter)

```
1 import tkinter
2 from tkinter import *
3 from tkinter.ttk import *

e = tkinter.Label(newwindow, width=24, text='First Name', borderwidth=2, relief='ridge', anchor='w', bg='yellow')
e.grid(row=0, column=0)
e = tkinter.Label(newwindow, width=24, text='Last Name', borderwidth=2, relief='ridge', anchor='w', bg='yellow')
e.grid(row=0, column=1)
e = tkinter.Label(newwindow, width=24, text='Date of Birth', borderwidth=2, relief='ridge', anchor='w', bg='yellow')
e.grid(row=0, column=2)
e = tkinter.Label(newwindow, width=24, text='Mobile Number', borderwidth=2, relief='ridge', anchor='w', bg='yellow')
e.grid(row=0, column=3)
e = tkinter.Label(newwindow, width=24, text='Email ID', borderwidth=2, relief='ridge', anchor='w', bg='yellow')
e.grid(row=0, column=4)
```

GUI used to import Tkinter and Tkinter form Tkinter . ttk import*

The includes the value new window, width, text, Boreder Width , Relicf ,anchor , Bg and gride include row and column.

The Tkinter used values (width= 24 , Borderwidth = 2,row=0,column=0 1 2 3 4)

The text used (First name , Last name ,Date of birth, Mobile number ,Email ID).

2. Implementation code (Python)

```
from datetime import datetime
from tkinter import messagebox

import psycopg2
import re
import datetime
from psycopg2 import Error
from psycopg2 import sql

window = Tk()
window.title('My Project')
window.geometry("600x400+10+20")

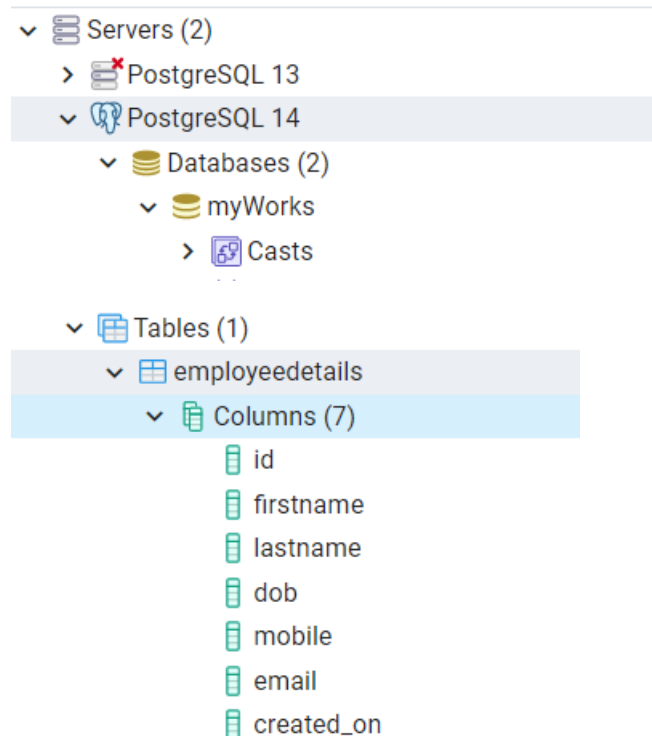
def opennewwindow():
    newwindow = Toplevel(window)
    newwindow.title("Employee Details")
    newwindow.geometry("900x600+10+20")
```

To implementation code used python is define the form datetime import datetime
next step include the Tkinter import Message box.

Mainly used import psycopg2 ,re, datetime, psycopg2 to import error and sql . MY
title used My Project and window . geometry ("600x400+10+20").

Def open window () make new window = Top level "Employee Details" new window
makes geometry ("900x600+10+20")

3. Database (PostgreSQL)



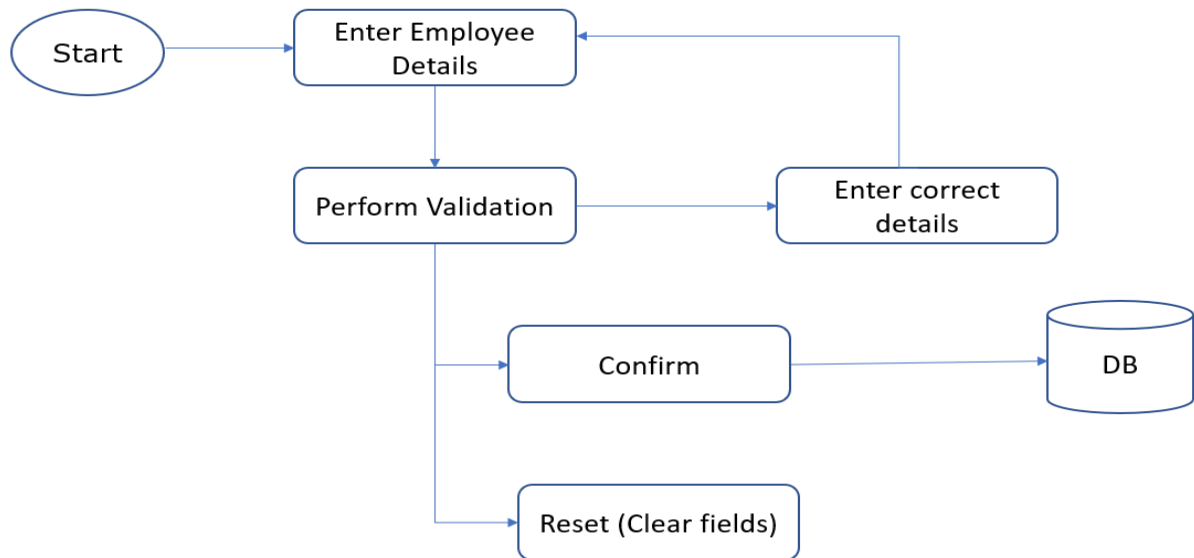
THE Database PostgreSQL 13, PostgreSQL 14

Tables (1) used employee details Columns (7) id, first name, last name, dob, mobile, email, created_on

CHAPTER 5

SYSTEM DESIGN

5.1 DATA FLOW DIAGRAM



CHAPTER 6

SNAP SHOTS

5.1.1 LEVEL 0

Employee on Boarding Tool to implemented Python

Install open-source tool **PyCharm**

Install **pgAdmin**

Install **PostgreSQL**

5.1.2 LEVEL 1

My Project

First Name

Last Name

Date of Birth

Mobile Number

Email ID

5.1.3 LEVEL 2

My Project

First Name

Last Name

Date of Birth

Mobile Number

Email ID

Can't NULL

 First Name, Last Name and Email can't be NULL

5.1.4 LEVEL 3

My Project

First Name

Last Name

Date of Birth

Mobile Number

Email ID

Insert Success

Record inserted successfully

Employee Details

First Name	Last Name	Date of Birth	Mobile Number	Email ID
SriRanga	Prasanth	04/09/2001	8975435335	sriranga@gmail.com

myWorks/postgres@PostgreSQL 14

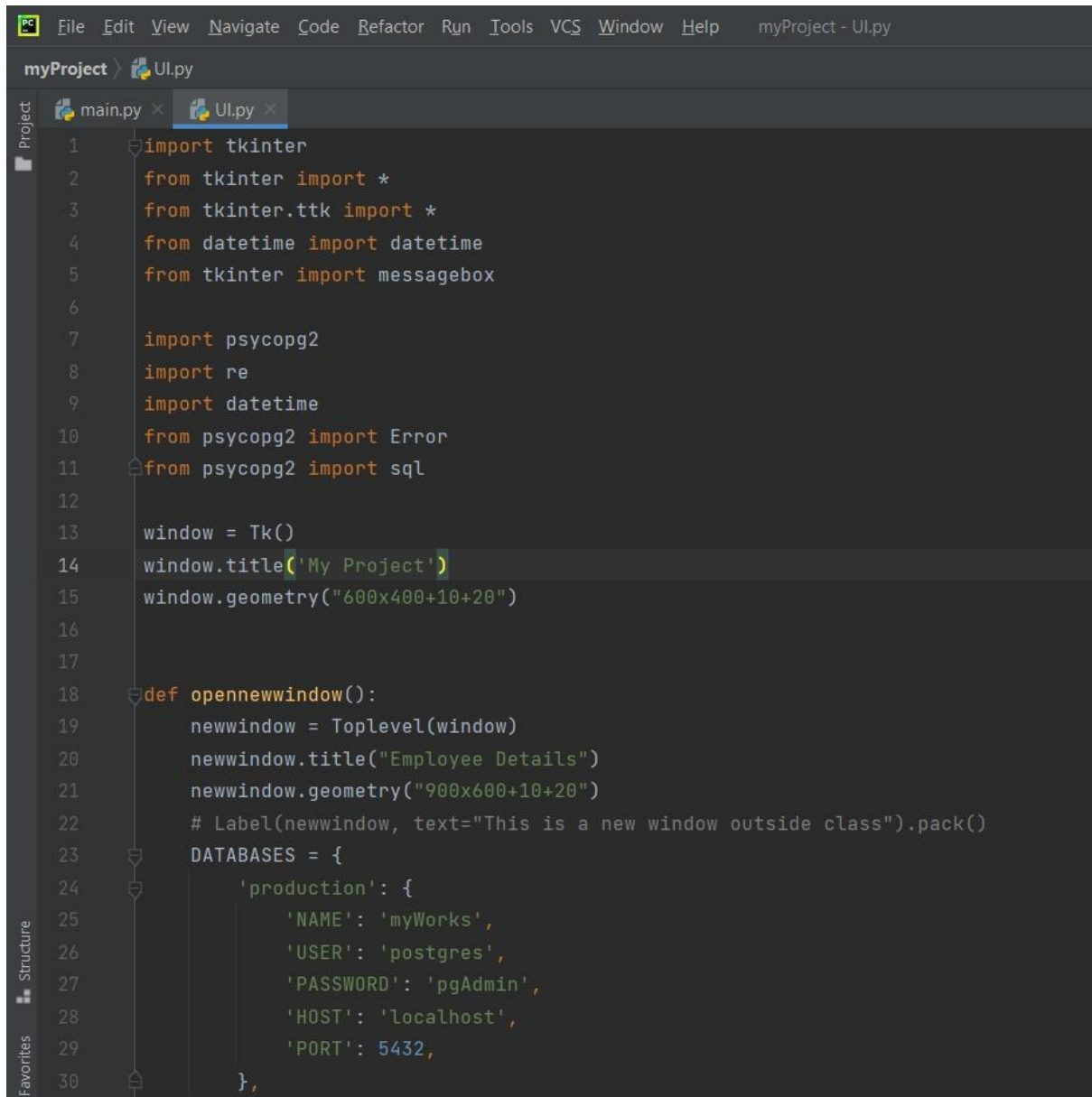
Query Editor Query History

```
1 select * from employeeDetails
```

Data Output Explain Messages Notifications

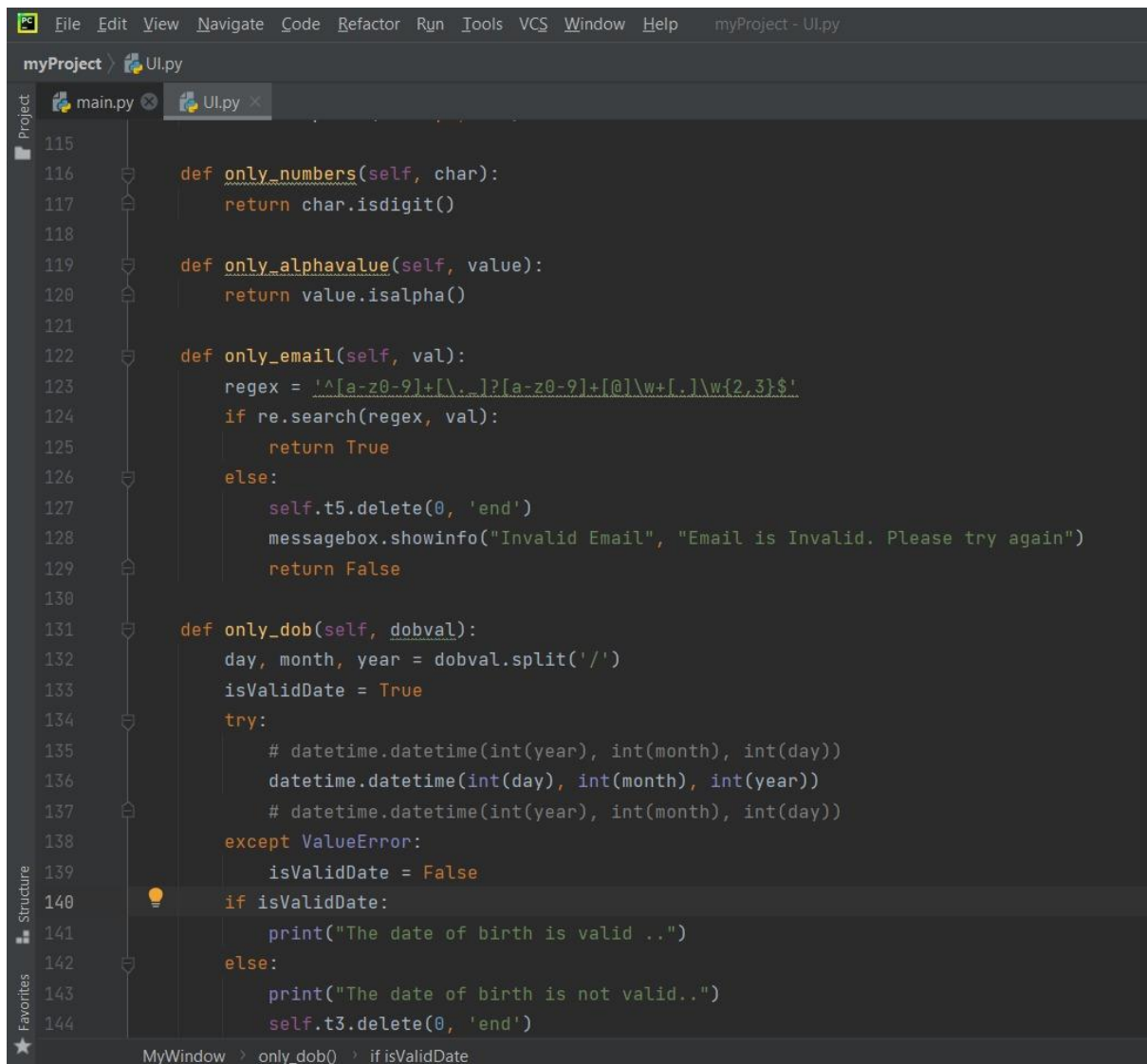
	id [PK] integer	firstname character varying (50)	lastname character varying (50)	dob character varying (50)	mobile character varying (50)	email character varying (255)	created_on timestamp without time zone
1	1	SriRanga	Prasanth	04/09/2001	8975435335	sriranga@gmail.com	2021-11-11 19:45:11.528654

SAMPLE CODING



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help myProject - Ul.py
myProject > Ul.py
main.py x Ul.py x
Project
1 import tkinter
2 from tkinter import *
3 from tkinter.ttk import *
4 from datetime import datetime
5 from tkinter import messagebox
6
7 import psycopg2
8 import re
9 import datetime
10 from psycopg2 import Error
11 from psycopg2 import sql
12
13 window = Tk()
14 window.title('My Project')
15 window.geometry("600x400+10+20")
16
17
18 def opennewwindow():
19     newwindow = Toplevel(window)
20     newwindow.title("Employee Details")
21     newwindow.geometry("900x600+10+20")
22     # Label(newwindow, text="This is a new window outside class").pack()
23     DATABASES = {
24         'production': {
25             'NAME': 'myWorks',
26             'USER': 'postgres',
27             'PASSWORD': 'pgAdmin',
28             'HOST': 'localhost',
29             'PORT': 5432,
30         },
```

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help myProject - UI.py
myProject UI.py
main.py x UI.py x
73
74 class MyWindow:
75     def __init__(self, win):
76         self.lbl1 = Label(win, text='First Name')
77         self.lbl2 = Label(win, text='Last Name')
78         self.lbl3 = Label(win, text='Date of Birth')
79         self.lbl4 = Label(win, text='Mobile Number')
80         self.lbl5 = Label(win, text='Email ID')
81         fnamevalidation = win.register(self.only_alphavalue)
82         self.t1 = Entry(win, validate="key", validatecommand=(fnamevalidation, '%S'))
83         self.t1.pack()
84         lnamevalidation = win.register(self.only_alphavalue)
85         self.t2 = Entry(win, validate="key", validatecommand=(lnamevalidation, '%S'))
86         self.t2.pack()
87         dobvalidation = win.register(self.only_dob)
88         self.t3 = Entry(win, validate="focusout", validatecommand=(dobvalidation, '%P'))
89         self.t3.pack()
90         # self.t4 = Entry(win, validate="key", validatecommand=(self.only_numbers, '%S'))
91         mobilevalidation = win.register(self.only_numbers)
92         self.t4 = Entry(win, validate="key", validatecommand=(mobilevalidation, '%S'))
93         self.t4.pack()
94         emailvalidation = win.register(self.only_email)
95         self.t5 = Entry(win, validate="focusout", validatecommand=(emailvalidation, '%P'))
96         self.t5.pack()
97         self.btn1 = Button(win, text='Confirm')
98         self.btn2 = Button(win, text='Reset')
99         self.lbl1.place(x=100, y=50)
100         self.t1.place(x=200, y=50, width=200)
101         self.lbl2.place(x=100, y=80)
102         self.t2.place(x=200, y=80, width=200)
MyWindow > _init_0
```



The screenshot shows an IDE window titled 'myProject - UI.py'. The code defines three validation methods: `only_numbers`, `only_alphavalue`, and `only_email`. The `only_email` method uses a regular expression to validate email addresses and shows an error message if invalid. The `only_dob` method (partially visible) splits a date string and attempts to create a datetime object to validate the date of birth.

```
115
116 def only_numbers(self, char):
117     return char.isdigit()
118
119 def only_alphavalue(self, value):
120     return value.isalpha()
121
122 def only_email(self, val):
123     regex = '^([a-z0-9]+[\.\_]?[a-z0-9]+@[a-z0-9]+\.[a-z]{2,3})$'
124     if re.search(regex, val):
125         return True
126     else:
127         self.t5.delete(0, 'end')
128         messagebox.showinfo("Invalid Email", "Email is Invalid. Please try again")
129         return False
130
131 def only_dob(self, dobval):
132     day, month, year = dobval.split('/')
133     isValidDate = True
134     try:
135         # datetime.datetime(int(year), int(month), int(day))
136         datetime.datetime(int(day), int(month), int(year))
137         # datetime.datetime(int(year), int(month), int(day))
138     except ValueError:
139         isValidDate = False
140     if isValidDate:
141         print("The date of birth is valid ..")
142     else:
143         print("The date of birth is not valid..")
144         self.t3.delete(0, 'end')
```

CONCLUSION

- Employee On Boarding tool is process successfully.
- Validation implemented for Name , Date of Birth ,Mobile Number and Email is process successfully.
- This project using to Employee on Boarding Tool.
- To implemented three modules.
- Employee details save successfully.

REFERENCES

- <https://www.programiz.com/python-programming>
- <https://realpython.com/python-gui-tkinter/>
- https://www.tutorialspoint.com/postgresql/postgresql_python.htm
- <https://www.postgresqltutorial.com/postgre>
e