DR. D. Y. PATIL INSTITUTE OF ENGINEERING, MANAGEMENT & RESEARCH, AKURDI

Department of Artificial Intelligence and Data Science

DEPARTMENT OF Artificial Intelligence and Data Science

317535: Internship

ACADEMIC YEAR 2023-24

“Python Development as a Jr. Software Developer”

NAME: NILKANTH DADAJI AHIRE

ROLL NO: TEAD21217

Department Of Artificial Intelligence and Data Science

DYP
DR. D. Y. PATIL
INSTITUTE OF
ENGINEERING,
MANAGEMENT
AND RESEARCH

DR. D. Y. PATIL INSTITUTE OF ENGINEERING, MANAGEMENT & RESEARCH, AKURDI

CERTIFICATE

This is to certify that the Internship Entitled

“Python Development as a Jr. Software Developer”

Submitted by

STUDENT NAME: - NILKANTH D. AHIRE **SEAT. NO: T191092004**

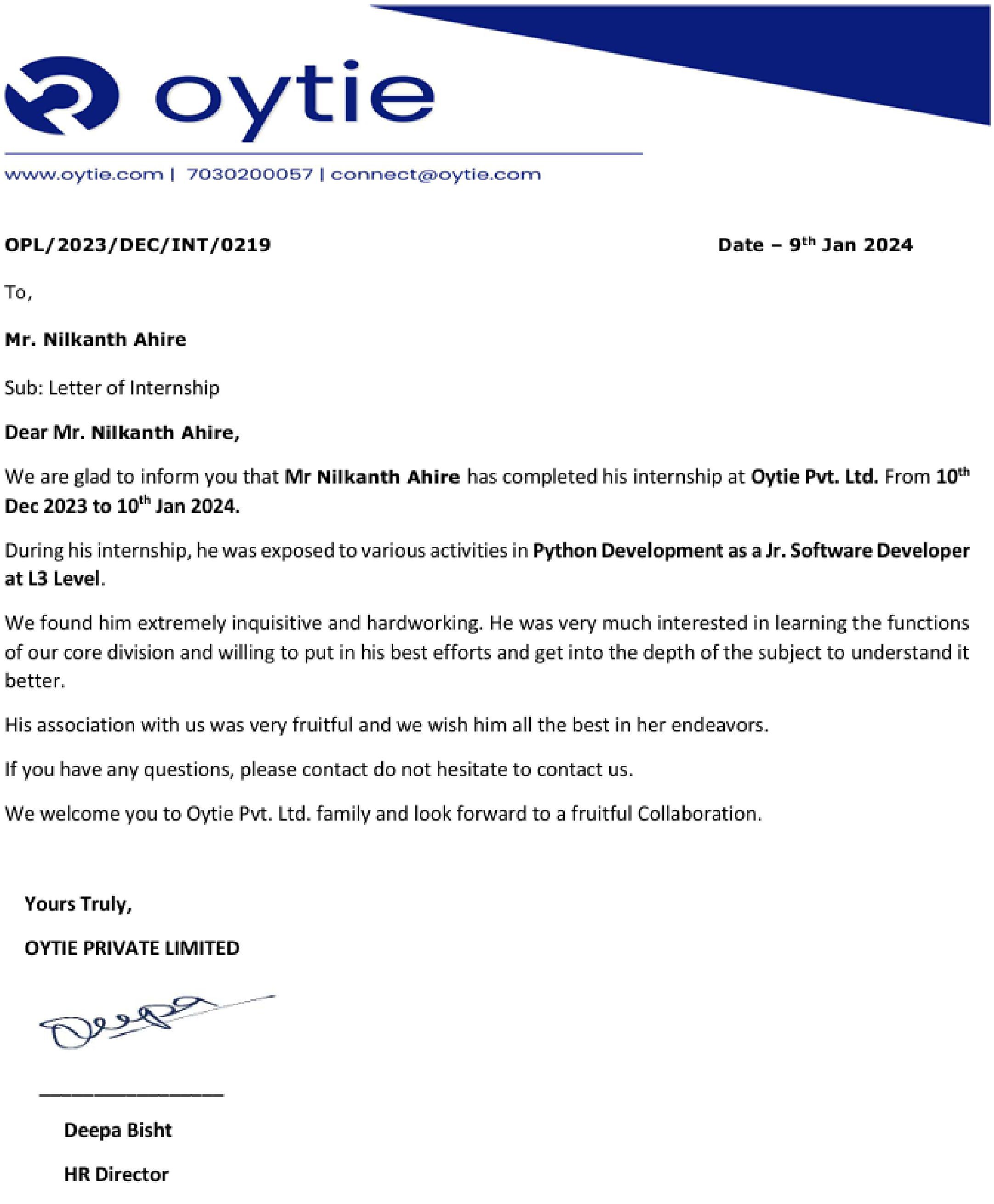
Dr. Suvarna Patil
H.O. D
Dept. of AIDS

Dr. Anupama V Patil
Principal
DYPIEMR

Signature of Internal Internship Supervisor

Signature of External Internship Supervisor

Internship Certificate



oytie

www.oytie.com | 7030200057 | connect@oytie.com

OPL/2023/DEC/INT/0219 **Date – 9th Jan 2024**

To,

Mr. Nilkanth Ahire

Sub: Letter of Internship

Dear Mr. Nilkanth Ahire,

We are glad to inform you that **Mr Nilkanth Ahire** has completed his internship at **Oytie Pvt. Ltd.** From **10th Dec 2023 to 10th Jan 2024.**

During his internship, he was exposed to various activities in **Python Development as a Jr. Software Developer at L3 Level**.

We found him extremely inquisitive and hardworking. He was very much interested in learning the functions of our core division and willing to put in his best efforts and get into the depth of the subject to understand it better.

His association with us was very fruitful and we wish him all the best in her endeavors.

If you have any questions, please contact do not hesitate to contact us.

We welcome you to Oytie Pvt. Ltd. family and look forward to a fruitful Collaboration.

Yours Truly,

OYTIE PRIVATE LIMITED

Deepa

____

Deepa Bisht

HR Director

Internship Place Details

Company background-Organization	Oytie
Activities/Scope	An organization which provides a platform for people to up-skill by getting practical experience throughout their career.
Objective of Study	• Learning web development • Improving Python programming skills • Gaining experience with database management • Understanding software development life cycle • Building a portfolio
Supervisor Details (Name, Designation, Company Name, Email- Id, Contact number)	• Mr. Sanjay Kharka • Operation Director • enquiry@oytie.com • **+917030200057**

Index

Introduction

In the dynamic realm of technology, theoretical knowledge forms the foundation, but practical experience serves as the cornerstone of expertise. As I reflect on my enriching journey through the internship in Python programming, I am compelled to encapsulate the myriad experiences, challenges, and achievements that have shaped my growth and proficiency in this versatile programming language.

1. Context and Objectives:

The internship in Python programming emerged as an opportunity to bridge the gap between academic learning and real-world application. With a solid foundation in programming principles and Python syntax acquired through coursework, the primary objective was to delve deeper into practical implementation, explore advanced concepts, and gain hands-on experience in developing real-world solutions.

2. Learning Outcomes:

The internship provided a fertile ground for personal and professional growth, fostering the development of a diverse skill set and a deeper understanding of Python's capabilities. Through immersive projects, collaborative endeavors, and mentorship from seasoned professionals, several key learning outcomes emerged:

Mastery of Python Language: Through daily coding exercises, project assignments, and code reviews, I honed my proficiency in Python, gaining fluency in syntax, data structures, object-oriented programming, and functional programming paradigms.
Application Development: I gained practical experience in developing a wide range of applications, including web development with frameworks like Django and Flask, desktop GUI applications using Tkinter, data analysis with libraries like Pandas and NumPy, and automation scripts for various tasks.
Problem-solving Skills: Tackling real-world challenges and debugging complex issues sharpened my problem-solving skills, teaching me to approach problems systematically, break them down into manageable components, and leverage Python's extensive ecosystem of libraries and tools.
Collaboration and Communication: Engaging in team projects and collaborative coding sessions facilitated the development of effective communication skills, teamwork, and the ability to collaborate seamlessly with colleagues to achieve common goals.

3. Projects and Contributions:

Throughout the internship, I had the opportunity to work on a diverse array of projects, each contributing to my growth and expanding my repertoire of skills:

Web Development Projects: Contributed to the development of dynamic web applications using Django and Flask frameworks, implementing features such as user authentication, database integration, RESTful APIs, and frontend design with HTML, CSS, and JavaScript.
GUI Application Development: Designed and developed desktop GUI applications using Tkinter, creating intuitive interfaces for tasks such as data visualization, file management, and user interaction.

Data Analysis and Visualization: Leveraged Python libraries such as Pandas, NumPy, and Matplotlib to perform data analysis, manipulate large datasets, extract insights, and visualize trends through graphs, charts, and interactive plots.
Automation Scripts: Developed automation scripts to streamline repetitive tasks, automate workflows, and improve efficiency in areas such as data processing, file management, system administration, and testing.

4. Challenges and Solutions:

The internship was not without its challenges, including learning curves, technical hurdles, and project constraints. However, each challenge presented an opportunity for growth and innovation. By seeking guidance, leveraging online resources, and adopting a growth mindset, I overcame obstacles, learned from setbacks, and emerged stronger and more resilient.

Title/Problem Statement/Objective

Title: -

Python Development as a Jr. Software Developer

Problem Statement: -

To develop competent and skilled web developers who can build dynamic and responsive web applications using Python, by providing hands-on training on the basics of web development, Python programming, database management, and deployment.

Objective: -

The primary objective of the internship is to provide students with hands-on experience in web development using Python. The course covers the following objectives:

- Understanding the fundamentals of Python programming language.
- Understanding the Database Management.
- Developing web applications using Python and Database Management.
- Deploying web applications to the web.

Motivation/Scope & Rationale of the Study

Motivation: -

Every journey begins with a single step, and in the realm of technology, every innovation starts with an idea. Today, we stand at the threshold of transforming the educational landscape through the power of Python programming. Imagine a world where managing student data isn't a daunting task but a seamless experience, where educators can harness the full potential of technology to enhance learning outcomes. This is where the vision of developing a Student Database Management System using Python comes into play.

Scope & Rationale of the study: -

Scope

- Learning the fundamentals of web development, such as HTML, CSS, and JavaScript.
- Learning the Python programming language.
- Becoming familiar with databases in web development, particularly SQL and ORM.
- Using Python and the Database Management, create responsive and dynamic web applications.
- Becoming familiar with the deployment and maintenance of web applications.

Rationale

- Individuals will gain hands-on experience in web programming using Python and the Database Management.
- To satisfy the growing need for experienced web developers across several sectors.
- To provide folks the chance to learn from industry leaders and seasoned web development professionals.
- To address the scarcity of qualified web developers by offering a forum for individuals to learn the skills and information required to pursue a career in web development.
- To provide people the opportunity to work on real-world projects and obtain hands-on experience in web development.
- To improve participants' employability and prepare them for a career in web development.

Daily Activity Report

- **Day 01 – 03: - 10 Dec – 12 Dec**
- **Topic Cover: - Introduction**
- **Summary: -**

  - At the start of the internship, they introduced us to the company and its domain. They outlined the services provided by the company, its mission, goals, the technologies utilized, and the projects currently being worked on.

  - After one day, they assigned a project to us in groups. We conducted requirement analysis and gained an understanding of the project. Subsequently, we commenced work on it.

  - Following is the Project Distribution:

Problem Statement: 'Student Database Management System'

- Part 1:

1. Create a web-app where a user can connect to database.
2. User can upload student details.
3. User can view his/her uploaded details.

- Part 2:

1. User can search and view details of students.
2. Users can delete all the records also.

- Additional Features:

1. Show date and time.
2. The user can connect to their own database and can add any number of student details and update them as well.

- Technologies Used:
  - Python
  - Tkinter
  - Python Database connectivity

- **Day 04 - 07: -12 Dec -16 Dec**
- **Topic Cover: - Python Tkinter-Add Student**
- **Summary: -**

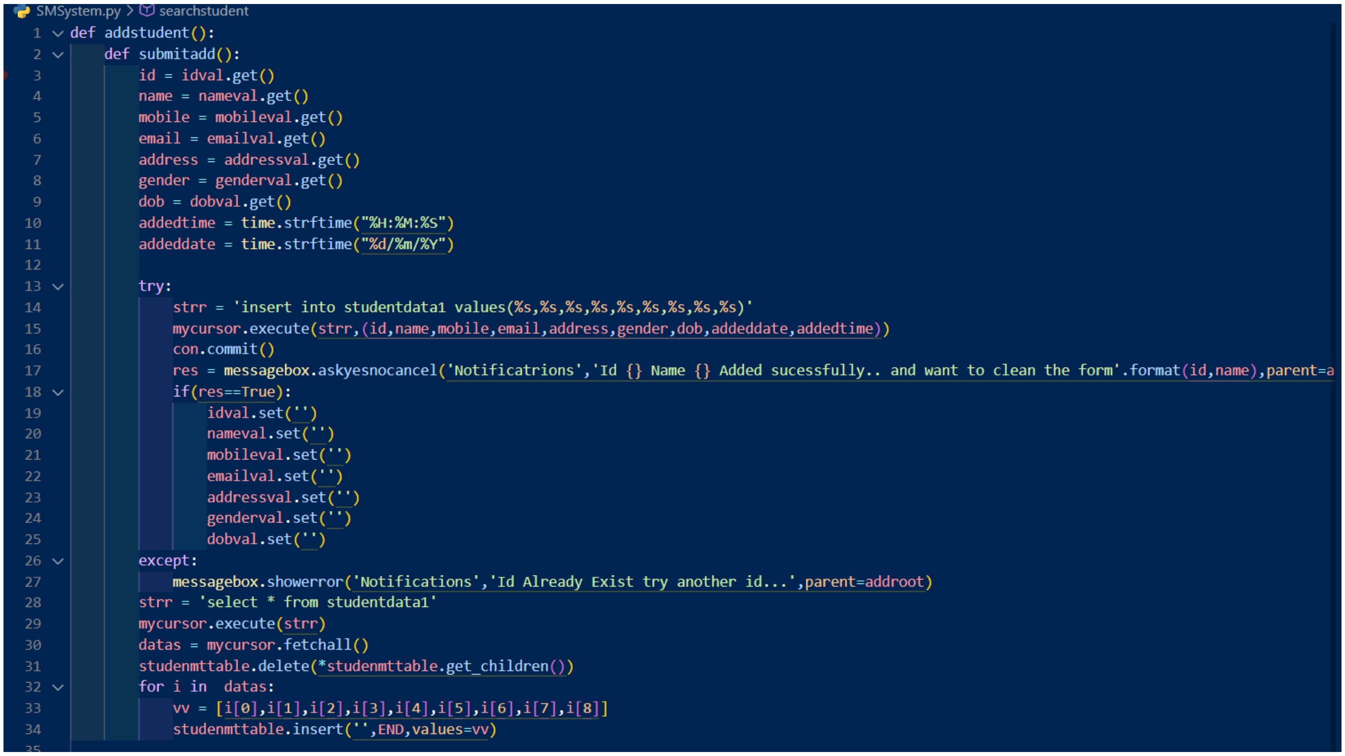
Tkinter is Python's standard GUI (Graphical User Interface) toolkit, bundled with most Python installations. It provides a fast and easy way to create desktop applications with graphical interfaces. Tkinter is based on the Tk GUI toolkit and is cross-platform, meaning it works on Windows, macOS, and Linux.

Key Features of Tkinter:

Simplicity: Tkinter is easy to learn and use, making it an excellent choice for beginners in GUI programming.
Cross-platform: Applications built with Tkinter work seamlessly across different operating systems without modification.
Rich Widget Set: Tkinter provides a wide range of widgets like buttons, labels, textboxes, menus, canvas, and more, enabling developers to create diverse GUI layouts.
Customizable: Tkinter allows for extensive customization of widgets, including their appearance, behavior, and layout.

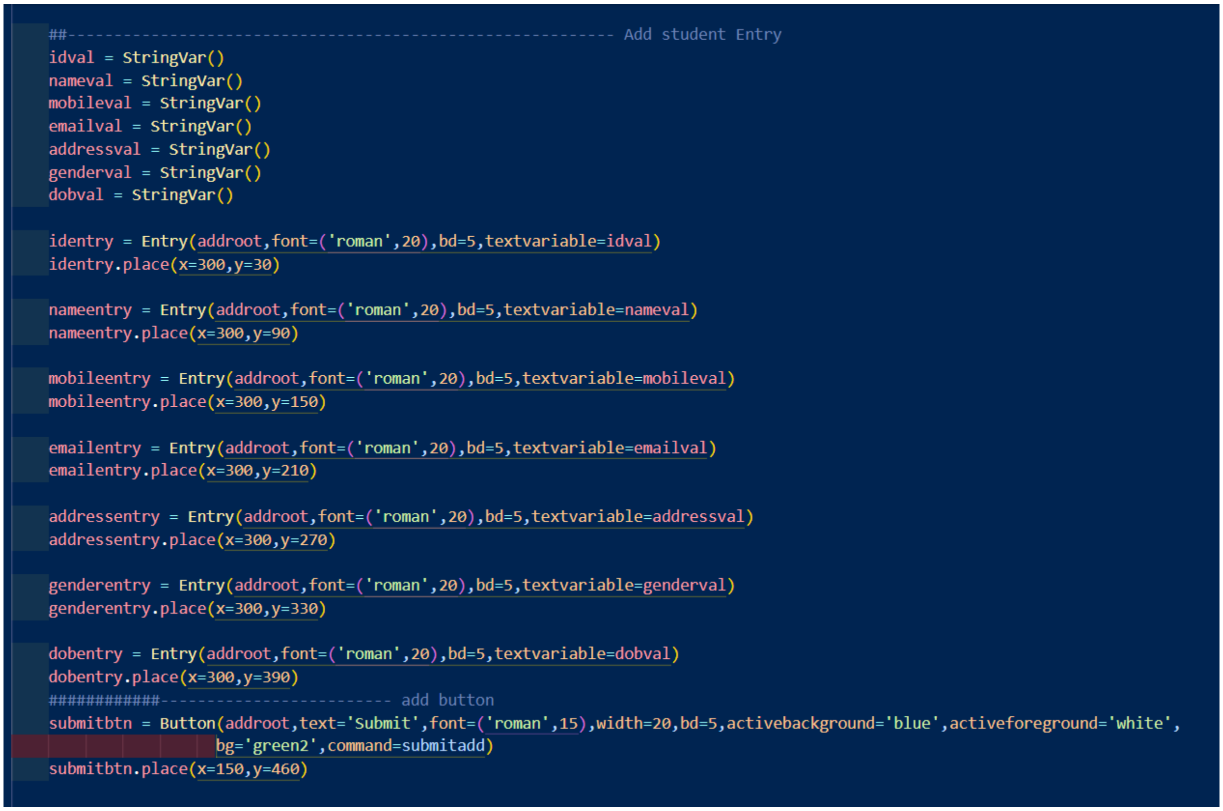
Usage in Python Projects:

Desktop Applications: Tkinter is commonly used to develop desktop applications with graphical interfaces. These applications range from simple utilities to complex software tools.
Data Visualization Tools: Tkinter can be used to create data visualization tools for analyzing and presenting data in a graphical format. It can integrate seamlessly with libraries like Matplotlib for advanced plotting capabilities.
Educational Projects: Tkinter is often used in educational projects to teach programming concepts and GUI development due to its simplicity and ease of use.
Utilities and Tools: Developers use Tkinter to create various utilities and tools, such as text editors, image viewers, calculators, and more.
Prototyping and Rapid Development: Tkinter is suitable for prototyping and rapid development of GUI applications, allowing developers to quickly iterate and test ideas.



```
def addstudent():
    def submitadd():
        id = idval.get()
        name = nameval.get()
        mobile = mobileval.get()
        email = emailval.get()
        address = addressval.get()
        gender = genderval.get()
        dob = dobval.get()
        addedtime = time.strftime("%H:%M:%S")
        addeddate = time.strftime("%d/%m/%Y")

        try:
            strr = 'insert into studentdata1 values(%s,%s,%s,%s,%s,%s,%s,%s,%s)'
            mycursor.execute(strr,(id,name,mobile,email,address,gender,dob,addeddate,addedtime))
            con.commit()
            res = messagebox.askyesnocancel('Notificatrions','Id {} Name {} Added sucessfully.. and want to clean the form'.format(id,name),parent=a
            if(res==True):
                idval.set('')
                nameval.set('')
                mobileval.set('')
                emailval.set('')
                addressval.set('')
                genderval.set('')
                dobval.set('')
        except:
            messagebox.showerror('Notifications','Id Already Exist try another id...',parent=addroot)
        strr = 'select * from studentdata1'
        mycursor.execute(strr)
        datas = mycursor.fetchall()
        studenmttable.delete(*studenmttable.get_children())
        for i in  datas:
            vv = [i[0],i[1],i[2],i[3],i[4],i[5],i[6],i[7],i[8]]
            studenmttable.insert('',END,values=vv)
```

- **Day 08 - 13: - 18 Dec – 23 Dec**
- **Topic Cover: - Student Entries**
- **Summary: -**



```
##---------------------------------------------------------- Add student Entry
idval = StringVar()
nameval = StringVar()
mobileval = StringVar()
emailval = StringVar()
addressval = StringVar()
genderval = StringVar()
dobval = StringVar()

identry = Entry(addroot,font=('roman',20),bd=5,textvariable=idval)
identry.place(x=300,y=30)

nameentry = Entry(addroot,font=('roman',20),bd=5,textvariable=nameval)
nameentry.place(x=300,y=90)

mobileentry = Entry(addroot,font=('roman',20),bd=5,textvariable=mobileval)
mobileentry.place(x=300,y=150)

emailentry = Entry(addroot,font=('roman',20),bd=5,textvariable=emailval)
emailentry.place(x=300,y=210)

addressentry = Entry(addroot,font=('roman',20),bd=5,textvariable=addressval)
addressentry.place(x=300,y=270)

genderentry = Entry(addroot,font=('roman',20),bd=5,textvariable=genderval)
genderentry.place(x=300,y=330)

dobentry = Entry(addroot,font=('roman',20),bd=5,textvariable=dobval)
dobentry.place(x=300,y=390)
############------------------------- add button
submitbtn = Button(addroot,text='Submit',font=('roman',15),width=20,bd=5,activebackground='blue',activeforeground='white',
                  bg='green2',command=submitadd)
submitbtn.place(x=150,y=460)
```

This code snippet appears to be part of a GUI (Graphical User Interface) application developed using Tkinter, a Python library for creating desktop applications. Let's break down the code step by step:

StringVar():

- This function is used to create an instance of the StringVar class, which is a special variable type in Tkinter designed to hold strings.
- In this snippet, several StringVar instances are created to store values entered by the user for different fields like ID, name, mobile number, email, address, gender, and date of birth.

Entry Widgets:

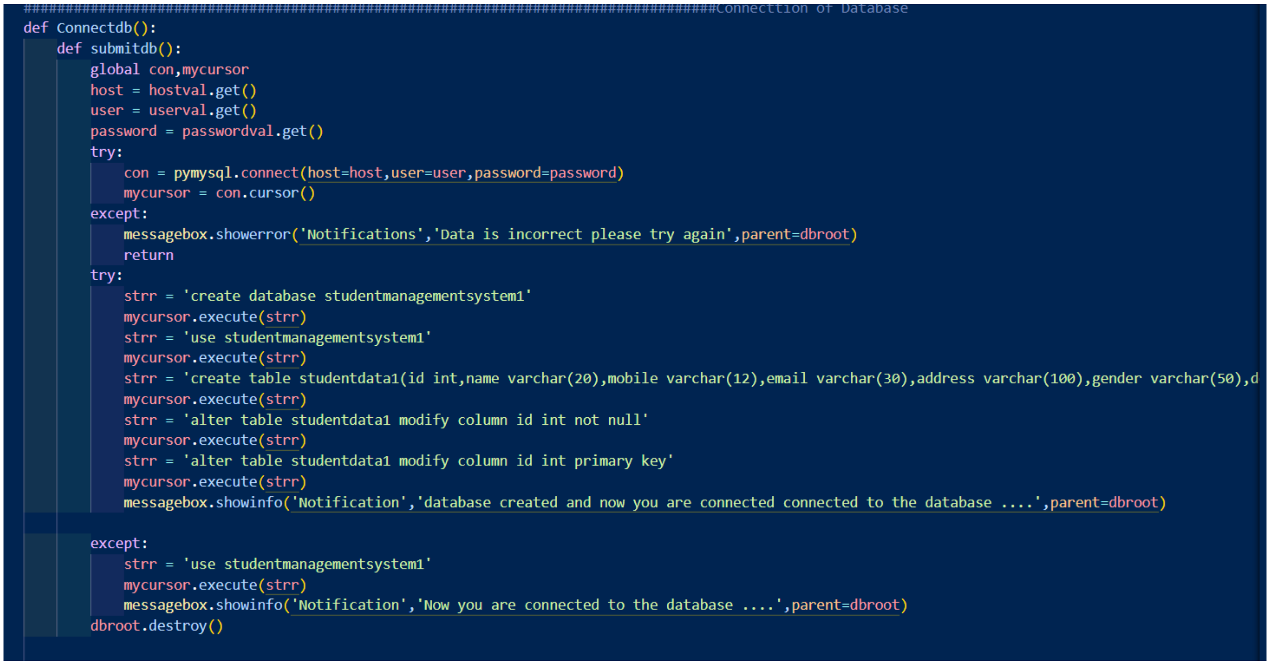
- Entry widgets are used to create text entry fields where users can input data.
- For each field (ID, name, mobile number, email, address, gender, and date of birth), an Entry widget is created with specific attributes:
- font=('roman',20): Sets the font style and size.

- bd=5: Sets the border width.
- textvariable: Associates the Entry widget with a StringVar instance, allowing bidirectional communication between the widget and the variable. Any changes made in the widget will update the associated variable, and vice versa.
- Each Entry widget is placed at a specific location within the GUI window using the place() method.

Button Widget:

- A Button widget is created with the text "Submit".
- Attributes such as font style, width, border width, background color, and foreground color are specified.
- The command parameter is set to a function submitadd, which presumably is a callback function that will be executed when the button is clicked.
- The Button widget is placed at a specific location within the GUI window using the place() method.

- **Day 14 - 15: - 25 Dec -26 Dec**
- **Topic Cover: - Database**
- **Summary: -**



```
###################################################################################Connecttion of Database
def Connectdb():
    def submitdb():
        global con,mycursor
        host = hostval.get()
        user = userval.get()
        password = passwordval.get()
        try:
            con = pymysql.connect(host=host,user=user,password=password)
            mycursor = con.cursor()
        except:
            messagebox.showerror('Notifications','Data is incorrect please try again',parent=dbroot)
            return
        try:
            strr = 'create database studentmanagementsystem1'
            mycursor.execute(strr)
            strr = 'use studentmanagementsystem1'
            mycursor.execute(strr)
            strr = 'create table studentdata1(id int,name varchar(20),mobile varchar(12),email varchar(30),address varchar(100),gender varchar(50),d
            mycursor.execute(strr)
            strr = 'alter table studentdata1 modify column id int not null'
            mycursor.execute(strr)
            strr = 'alter table studentdata1 modify column id int primary key'
            mycursor.execute(strr)
            messagebox.showinfo('Notification','database created and now you are connected connected to the database ....',parent=dbroot)

        except:
            strr = 'use studentmanagementsystem1'
            mycursor.execute(strr)
            messagebox.showinfo('Notification','Now you are connected to the database ....',parent=dbroot)
        dbroot.destroy()
```

Database Connection:

- The function retrieves the host, username, and password values entered by the user from variables hostval, userval, and passwordval, respectively.
- It then attempts to establish a connection to the MySQL database using the pymysql.connect() function with the provided credentials.
- If the connection is successful, it sets the global variables con and mycursor to the connection object and cursor, respectively. Otherwise, it displays an error message using messagebox.showerror()

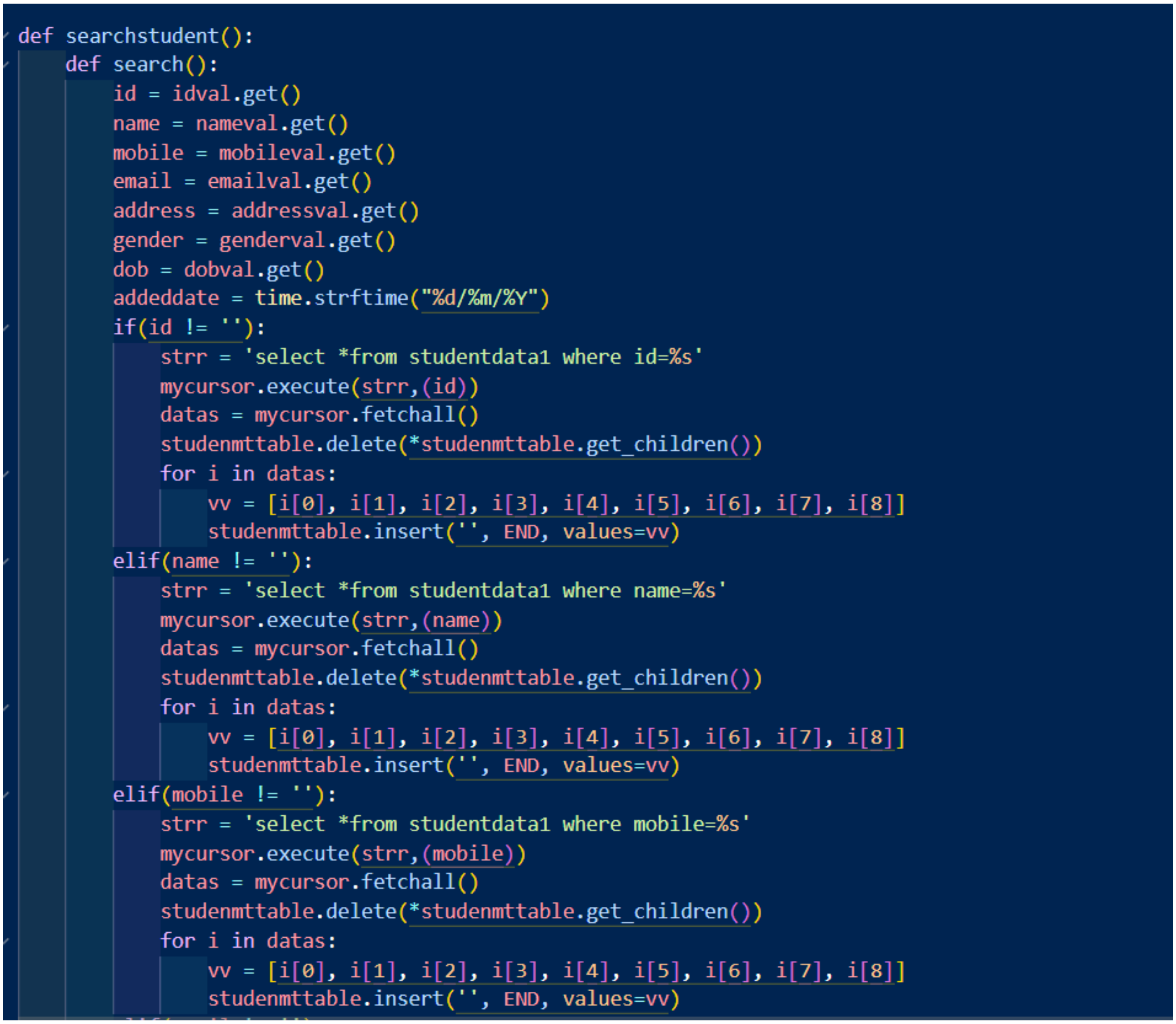
Database Schema Creation:

- After establishing the connection, the function attempts to create a new database named 'studentmanagementsystem1' using the SQL statement: CREATE DATABASE studentmanagementsystem1.
- It then switches to the newly created database using the SQL statement: USE studentmanagementsystem1.
- Next, it creates a table named 'studentdata1' within the database, defining columns for student details such as ID, name, mobile number, email, address, gender, date of birth, date, and time.
- The function modifies the 'id' column of the 'studentdata1' table to be not null and sets it as the primary key.
- If any of these operations encounter an exception, it catches the exception, switches to the existing

database 'studentmanagementsystem1' (if already created), and displays an appropriate message.

Overall, this function encapsulates the logic for connecting to a MySQL database, creating a database schema, and handling exceptions gracefully. It provides feedback to the user through message notifications, ensuring a smooth user experience.

- **Day 16 - 20: - 27 Dec – 30 Dec**
- **Topic Cover: - Search Student**
- **Summary: -**



```
def searchstudent():
    def search():
        id = idval.get()
        name = nameval.get()
        mobile = mobileval.get()
        email = emailval.get()
        address = addressval.get()
        gender = genderval.get()
        dob = dobval.get()
        addeddate = time.strftime("%d/%m/%Y")
        if(id != ''):
            strr = 'select *from studentdata1 where id=%s'
            mycursor.execute(strr,(id))
            datas = mycursor.fetchall()
            studenmttable.delete(*studenmttable.get_children())
            for i in datas:
                vv = [i[0], i[1], i[2], i[3], i[4], i[5], i[6], i[7], i[8]]
                studenmttable.insert('', END, values=vv)
        elif(name != ''):
            strr = 'select *from studentdata1 where name=%s'
            mycursor.execute(strr,(name))
            datas = mycursor.fetchall()
            studenmttable.delete(*studenmttable.get_children())
            for i in datas:
                vv = [i[0], i[1], i[2], i[3], i[4], i[5], i[6], i[7], i[8]]
                studenmttable.insert('', END, values=vv)
        elif(mobile != ''):
            strr = 'select *from studentdata1 where mobile=%s'
            mycursor.execute(strr,(mobile))
            datas = mycursor.fetchall()
            studenmttable.delete(*studenmttable.get_children())
            for i in datas:
                vv = [i[0], i[1], i[2], i[3], i[4], i[5], i[6], i[7], i[8]]
                studenmttable.insert('', END, values=vv)
```

Nested Function search():

- Inside the searchstudent() function, there's a nested function named search().
- This nested function is presumably intended to be called when the user performs a search operation.
- Retrieving Input Values:
- The function retrieves the values entered by the user for various search criteria, such as ID, name, mobile number, email, address, gender, date of birth, and added date.
- These values are obtained from the corresponding StringVar variables (idval, nameval, etc.).

Performing Database Queries:

- Based on the search criteria provided by the user, the function constructs SQL queries to retrieve matching records from the database table studentdata1.
- It dynamically generates the SQL query based on the non-empty search criteria provided by the user.
- For each search criteria, a separate SQL query is constructed and executed using mycursor.execute().
- The fetched data is stored in the variable datas.

Populating Table with Search Results:

- After executing the query, the function clears any existing data in the table (studenttable) using studenttable.delete().
- It then iterates through the fetched data (datas) and inserts each record into the table as a new row.
- The table is populated with the search results, allowing the user to view the matching records.

Creating Search Interface:

- The function creates a new Toplevel window (searchroot) to serve as the search interface.
- It sets the window title, dimensions, background color, icon, and prevents resizing.
- The search interface likely includes input fields for specifying search criteria and a button to initiate the search operation.

- **Day 21-25: - 1 Jan – 5 Jan**
- **Topic Cover: - Data Entry Frame**
- **Summary: -**

Frame Creation:

- A Frame widget named DataEntryFrame is created with specific attributes such as background color (bg), relief style (relief), and border width (borderwidth).
- The Frame is placed at coordinates (x=10, y=80) within the root window, with a width of 500 and height of 600 pixels.

Front Label:

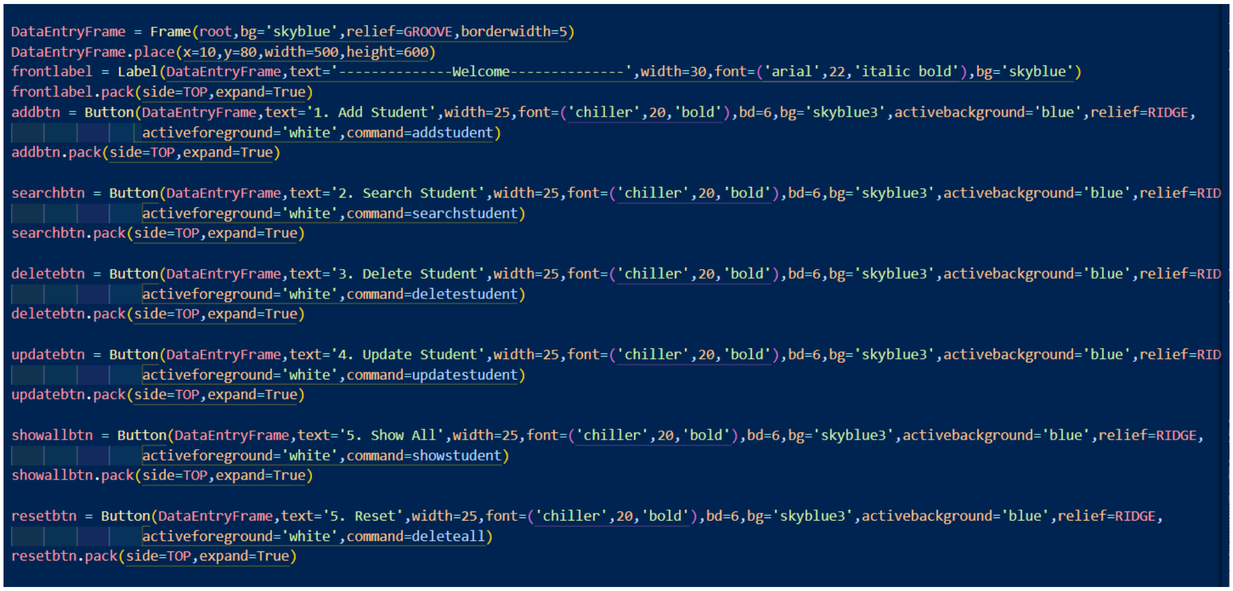
- A Label widget named frontlabel is created within the DataEntryFrame.
- It displays the text "--------------Welcome--------------" with specific attributes such as font style (font) and background color (bg).
- The label is positioned at the top (side=TOP) of the DataEntryFrame and set to expand (expand=True) to fill the available space horizontally.

Operation Buttons:

- Five Button widgets (addbtn, searchbtn, deletebtn, updatebtn, showallbtn, and resetbtn) are created within the DataEntryFrame to perform different operations.
- Each button is labeled with a specific operation (e.g., "Add Student", "Search Student", etc.).
- Buttons have similar attributes such as width, font style, border width (bd), background color (bg), relief style (relief), and active background/foreground colors (activebackground, activeforeground).
- Each button is associated with a command callback function (command) that will be executed when the button is clicked. For example, addstudent, searchstudent, etc. are callback functions defined elsewhere in the code.

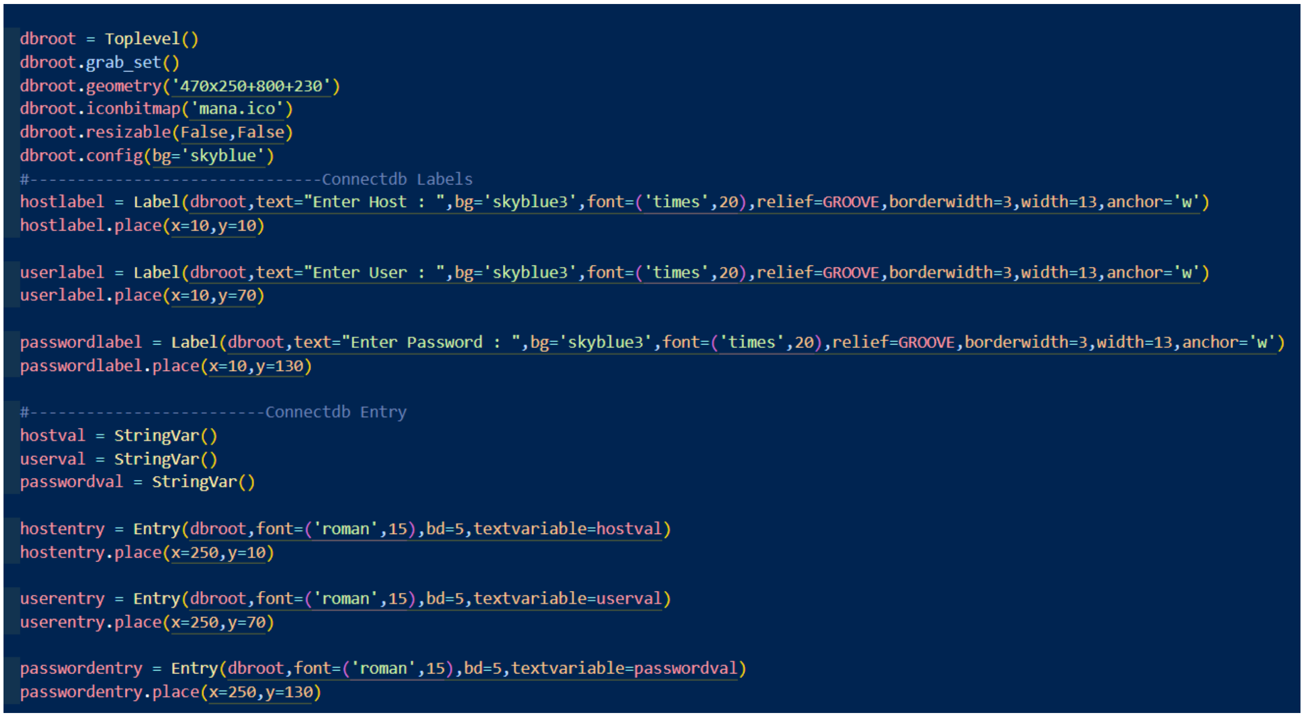
Button Packing:

- Each button is packed within the DataEntryFrame using the pack() method with specific packing options (e.g., side=TOP, expand=True) to control their placement and behavior within the frame.



```
DataEntryFrame = Frame(root,bg='skyblue',relief=GROOVE,borderwidth=5)
DataEntryFrame.place(x=10,y=80,width=500,height=600)
frontlabel = Label(DataEntryFrame,text='--------------Welcome--------------',width=30,font=('arial',22,'italic bold'),bg='skyblue')
frontlabel.pack(side=TOP,expand=True)
addbtn = Button(DataEntryFrame,text='1. Add Student',width=25,font=('chiller',20,'bold'),bd=6,bg='skyblue3',activebackground='blue',relief=RIDGE,
                activeforeground='white',command=addstudent)
addbtn.pack(side=TOP,expand=True)

searchbtn = Button(DataEntryFrame,text='2. Search Student',width=25,font=('chiller',20,'bold'),bd=6,bg='skyblue3',activebackground='blue',relief=RID
                activeforeground='white',command=searchstudent)
searchbtn.pack(side=TOP,expand=True)

deletebtn = Button(DataEntryFrame,text='3. Delete Student',width=25,font=('chiller',20,'bold'),bd=6,bg='skyblue3',activebackground='blue',relief=RID
                activeforeground='white',command=deletestudent)
deletebtn.pack(side=TOP,expand=True)

updatebtn = Button(DataEntryFrame,text='4. Update Student',width=25,font=('chiller',20,'bold'),bd=6,bg='skyblue3',activebackground='blue',relief=RID
                activeforeground='white',command=updatestudent)
updatebtn.pack(side=TOP,expand=True)

showallbtn = Button(DataEntryFrame,text='5. Show All',width=25,font=('chiller',20,'bold'),bd=6,bg='skyblue3',activebackground='blue',relief=RIDGE,
                activeforeground='white',command=showstudent)
showallbtn.pack(side=TOP,expand=True)

resetbtn = Button(DataEntryFrame,text='5. Reset',width=25,font=('chiller',20,'bold'),bd=6,bg='skyblue3',activebackground='blue',relief=RIDGE,
                activeforeground='white',command=deleteall)
resetbtn.pack(side=TOP,expand=True)
```

- **Day 26 - 30: - 6 Jan – 10 Jan**
- **Topic Cover: - Database Frame.**
- **Summary: -**



```
dbroot = Toplevel()
dbroot.grab_set()
dbroot.geometry('470x250+800+230')
dbroot.iconbitmap('mana.ico')
dbroot.resizable(False,False)
dbroot.config(bg='skyblue')
#-------------------------------Connectdb Labels
hostlabel = Label(dbroot,text="Enter Host : ",bg='skyblue3',font=('times',20),relief=GROOVE,borderwidth=3,width=13,anchor='w')
hostlabel.place(x=10,y=10)

userlabel = Label(dbroot,text="Enter User : ",bg='skyblue3',font=('times',20),relief=GROOVE,borderwidth=3,width=13,anchor='w')
userlabel.place(x=10,y=70)

passwordlabel = Label(dbroot,text="Enter Password : ",bg='skyblue3',font=('times',20),relief=GROOVE,borderwidth=3,width=13,anchor='w')
passwordlabel.place(x=10,y=130)

#-------------------------Connectdb Entry
hostval = StringVar()
userval = StringVar()
passwordval = StringVar()

hostentry = Entry(dbroot,font=('roman',15),bd=5,textvariable=hostval)
hostentry.place(x=250,y=10)

userentry = Entry(dbroot,font=('roman',15),bd=5,textvariable=userval)
userentry.place(x=250,y=70)

passwordentry = Entry(dbroot,font=('roman',15),bd=5,textvariable=passwordval)
passwordentry.place(x=250,y=130)
```

Toplevel Creation:

- A new Toplevel widget named dbroot is created to serve as a pop-up window for database connection details.
- The grab_set() method ensures that this pop-up window gains focus and blocks interactions with other windows.
- Specific attributes such as geometry (size and position), icon, and background color are configured for the Toplevel window.

Labels for Connection Details:

- Three Label widgets (hostlabel, userlabel, passwordlabel) are created to prompt the user to enter host, username, and password, respectively.
- Each label specifies the text to be displayed, background color, font style, relief style (relief), border width (borderwidth), width, and text alignment (anchor).
- Labels are positioned at specific coordinates within the dbroot window using the place() method.

Methodological Details

- **Needs Assessment:** The program begins with a needs assessment to determine the participants' learning requirements and objectives. A pre-internship survey and individual interviews with participants are used to determine their expectations, aspirations, and past knowledge.

- **Practical tasks:**

  The program includes a variety of practical tasks that allow participants to apply their knowledge to real-world situations. This assignment is designed to reinforce theoretical concepts and provide practical experience with tools such as Python, NumPy, Pandas, and Tkinter. Depending on the task, participants may work individually or collaboratively, allowing them to develop their skills in both settings.

- **Hands-on Projects:** The program includes hands-on project that provide learners the opportunity to work on real-world Python project. The project is meant to imitate real- world problems and allow participants to apply their knowledge in a practical environment. Depending on the nature of the project, project is performed either alone or in groups.

- **Mentorship & Support:** Throughout the internship, participants will get mentorship and support from the program. Each participant is allocated a mentor who will provide direction, support, and feedback on their assignments and project. Participants can also ask trainers and mentors for help via email, online forums, or one-on-one meetings.

- **Assessment and Evaluation:** The program includes evaluations to track learning outcomes and internship effectiveness. They use to daily keep track of the project and task given and track their completion as well.

Result/Analysis/Conclusion

Result

The Student Management System (SMS) developed using Python Tkinter and MySQL database connectivity has been successfully implemented, offering a comprehensive solution for managing student information in educational institutions. The project encompasses various features, including adding, searching, deleting, updating, and displaying student records through an intuitive graphical user interface. Functionally, the SMS provides essential capabilities crucial for efficient student management. Users can add new student records, search for specific students based on various criteria, update existing records, delete outdated entries, and view all student records seamlessly through the user-friendly interface. The project establishes a robust connection to a MySQL database, enabling the storage and retrieval of student data. CRUD operations (Create, Read, Update, Delete) are performed effectively, ensuring data integrity and reliability. The integration of database connectivity enhances the system's capability to manage large volumes of student information efficiently. The Tkinter-based GUI provides a user-friendly interface with clear navigation options and input fields for data entry. Buttons and labels are strategically placed to enhance usability, making it easy for users to navigate through different functionalities. The interface design prioritizes simplicity and intuitiveness, contributing to a positive user experience. Robust error handling mechanisms are incorporated into the SMS to ensure smooth operation. Error messages guide users to correct input errors, and exception handling prevents application crashes, ensuring uninterrupted usage.

Analysis

The development of the Student Management System marks a significant achievement in educational administration, offering a modern and efficient solution for managing student data. The project's success can be attributed to several factors. Firstly, the effective functionality of the SMS fulfills its intended purpose by providing essential functionalities for student management. The system's reliable database connectivity ensures the storage and retrieval of student data, effectively handling CRUD operations while maintaining data integrity and scalability. Additionally, the user-friendly interface of the Tkinter-based GUI enhances usability, allowing users to navigate and interact with the system effortlessly. Clear labeling and intuitive design contribute to a positive user experience, enabling users to perform tasks efficiently. Lastly, the robust error handling mechanisms implemented in the SMS guide users to correct input errors and prevent application crashes, ensuring the overall reliability of the system. Overall, the Student Management System represents a successful implementation of Python programming principles and database connectivity in the context of educational administration, providing a valuable tool for educational institutions seeking to optimize student management practices.

Conclusion

The Student Management System represents a successful implementation of Python programming principles and database connectivity in the context of educational administration, providing a valuable tool for educational institutions seeking to optimize student management practices. The project's effective functionality, reliable database connectivity, user-friendly interface, and robust error handling mechanisms contribute to its overall success. Moving forward, the SMS can be further enhanced with additional features and refinements to meet evolving requirements and address specific needs of educational institutions. Overall, the SMS serves as a testament to the power of technology in streamlining administrative processes and enhancing data management in educationalsettings.

Suggestions

- **Establish Specific Goals and Objectives:** Establish specific goals and objectives for the internship program and convey them to participants. This will assist students in understanding what is expected of them and what they may anticipate from the program.

- **Provide a Planned Curriculum:** The Python Programming Internship curriculum covers a comprehensive range of topics, beginning with the fundamentals of Python programming and progressing to advanced concepts in data science. Participants will learn data manipulation, statistical analysis, and data visualization techniques using Python libraries such as NumPy, Pandas, Matplotlib, and Seaborn.

- **Encourage Collaboration:** Encourage people to work on projects and activities together. This will allow them to learn from one another, share ideas, and improve their collaboration abilities.

- **Provide Feedback Opportunities:** Give participants regular feedback on their progress and constructive criticism to help them develop their abilities. This will keep them motivated and focused on their objectives.

- **Mentorship:** Provide mentorship to participants from experienced individuals in the field of web development. This will give students with career counselling, assistance, and industry insights that will help them succeed.

- **Encourage Networking:** Encourage participants to network with other web development experts. Attending business events, participating in online forums, and engaging in social media activities are all ways to do this.

Attendance Record

During the internship program, attendance was taken at each mentor meeting, which occurred on weekends. The attendance report shows that the candidate attended all mentor meetings and completed all assigned tasks. However, **attendance was not taken on a daily basis** for each module as the program was designed to be self-paced and flexible to accommodate the individual needs of each participant.

It is important to note that **completion of all modules was a requirement for participation in the project** component of the internship program. Therefore, **the candidate completed all modules and was eligible for the project**. The project was assigned to the candidate on 2023-12-13, and the submission deadline was on 2024-01-09.

Sr. No.	Date	Company Supervisor Sign
1.	10-12-2023	Deepa
2.	15-12-2023	Deepa
3.	18-12-2023	Deepa
4.	22-12-2023	Deepa
5.	26-12-2023	Deepa
6.	01-01-2024	Deepa
7.	04-01-2024	Deepa
8.	07-01-2024	Deepa

Deepa

Lead Instructor

Acknowledgment

It is our pleasure to acknowledge sense of gratitude to all those who helped us in completion of Internship. I am highly indebted to **Miss. Kalyani Kute** from DYPIEMR, Akurdi & **Mr. Sanjay Kharka** for their guidance and constant supervision as well as for providing necessary information regarding the internship & also for their support.

I would like to express my gratitude towards **Dr. Suvarna Patil (Head - Department of AIDS)** for their kind co-operation and encouragement which help me for providing the required facilities.

Finally, we wish to thank and appreciate to all our teachers and friends for their constructive comments, suggestions and guidance and all those directly or indirectly helped us in completing this internship.

List of References

1. https://www.python.org/documentations/
2. https://realpython.com/tutorials/web-dev/
3. Python for Software Development.pdf (halvorsen.blog)
4. Learn Python - Free Interactive Python Tutorial
5. The Python Language Reference — Python 3.12.3 documentation

Evaluation Sheet

Sr No.	Evaluation Parameters	Marks
1	Attendance Record (Out of 25)	
2	Diary/Workbook (Out of 25)	

Signature of Internal Internship Supervisor

Signature of External Internship Supervisor