AIRCRAFT MANAGEMENT SYSTEM

MINOR PROJECT REPORT

By
DHRUV SHARMA (RA2211003011548)
KARAN RAJ SHARMA (RA2211003011550)
SIDHARTH M BABU (RA2211003011608)

Under the guidance of

DR. ASHWINI S

Assistant Professor, Department of Computing Technologies

In partial fulfilment for the Course

of

21CSC203P –ADVANCED PROGRAMMING PRACTICE

in Department of Computing Technologies

FACULTY OF ENGINEERING AND TECHNOLOGY

SCHOOL OF COMPUTING

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR

NOVEMBER 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this minorproject report for the course **21CSC203P ADVANCED PROGRAMMING PRACTICE** entitled in " **Aircraft Management System**" is the bonafide work of **Dhruv Sharma (RA2211003011548), Karan Raj Sharma (RA2211003011550) and Sidharth M Babu (RA2211003011608)** who carried out the work under my supervision.

SIGNATURE

Dr. Ashwini S

Assistant Professor

Department of Computing Technologies

SRM Institute of ScienceandTechnology

Kattankulathur

ABSTRACT

The Aircraft Management System (AMS) project is aimed at developing a comprehensive software platform for managing aircraft fleets. The system will be designed using the Python programming language, and it will consist of two primary components, a Graphical User Interface (GUI), and an Application Programming Interface (API). The GUI will provide an interface for users to interact with the system, while the API will provide a back-end interface for the system, allowing it to connect to external systems and databases. The AMS will enable airline companies to manage their fleets more efficiently, reducing costs associated with maintenance, fuel, and crew scheduling while increasing efficiency and revenue. By using the Agile methodology, the AMS project will be developed in sprints, with continuous feedback and testing, ensuring that the AMS meets the requirements and is user-friendly.

ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all ourendeavors.

We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy,** for his encouragement.

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V.Gopal,** for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman,** for imparting confidence to complete my courseproject.

We wish to express my sincere thanks to **Course Audit Professors Dr. Vadivu. G, Professor, Department of Data Science and Business Systems and Dr. Sasikala. E Professor, Department of Data Science and Business Systems** and **Course Coordinators** for their constant encouragement and support.

We are highly thankful to our my Course project Faculty **Dr. Ashwini S, Assistant Professor, Department of Computing Technologies** for her assistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our **HoD Dr. M. Pushpalatha , Professor & Head , Department of Computing Technologies** and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

TABLE OF CONTENTS

1. INTRODUCTION

Our project was created with one common thing in mind that we want the aviation industry (mainly areas like data control center etc.,.) to have a change from their usual point of using pen and papers to change them into online and internet which was necessary as there are many problems with the current strategy of pen and paper. So, we started my research on how thecurrent system of paper and pen work then we defined some new systems.

We modified the issues from the pen paper systems to fit for our systems by including the following things:

1) Introduced a new GUI (Graphical User Interface) to make it easy and to enter, delete, edit and retrieve data from the tables in an attractive manner.

2) Used python as my front end because python is the only language which is one of the developed, known, understandable and easier programming language for the current generation.

3) Used SQL as my back end because SQL is the evergreen, basic, best and organized software for storing the information

.It also has well defined parameters ,the userscan retrieve the details easily from SQL by the specified syntax and also this is the software which shows best result when integrated with python.

In this project of 'AIRCRAFT SCOUT', the windows that had been used is actually created by the python module named 'Tkinter' .This module is actually a graphical module(GUI) in python which provided many exciting features such as message box generation ,Window creation, Buttons ,labels and so on to complete and design this project.

This project has a wide range of applications especially in Flight systems. The main objective of this project is to reduce the problems of storing and accessing the flight's and pilot's data. This project actually helps in inserting, fetching, updating and deleting the pilot and flight details in fraction of seconds. They are more preferable for storing information and also many advantages such as reliability and safety assurance. They are also accurate. They can only be accessed by working members of company. The details are protected by strong passwords and thus, this project provides good authentication facilities too.

2. LITERATURE SURVEY

There are various aircraft management systems available in the market, such as AeroCRS, AVB, and AMOS. These systems are designed to handle complex airline operations, including maintenance, crew scheduling, and revenue management. However, many of these systems are expensive and complex, making them less accessible to small and mid- size airlines. Therefore, this project will develop an AMS that is cost-effective, user- friendly, and designed to meet the requirements of small and mid-size airlines. The literature survey will review current aircraft management systems, their features, and their limitations. This will help identify areas where the proposed AMS can be improved to provide better services to airline companies.

This project is made to modernize the way we maintain records. Python is used as front endand SQL (Structured Query Language).The authentication had been done in such a way that, unauthorised users or strangers could not access the system. In this system, the user can Select, Insert, Delete and update the Flight and Pilot details in this project.

Advantages:

1.Safer than paper ones as they can't be easily destroyed by natural or man- made calamities.

2. The data present in the digital form cannot be deleted/damaged without leaving any traces so hackers/crackers will be easily caught.

3. Easily accessible as you don't need to look from pages to pages searching for one particular row, rather than use the search function, so time saved is time gained.

4. Transfer of data/transfer rate is more than its counter parts like paper, so instead of waiting days for the paper ones to reach you, you can receive it in less than 5minutes through Google mail/Google drive.

5. Storage space is so low that we can store 1 terra byte of data of flights and its records which take space compared to our palm while the same size of data when in paper forms will take space of about a big room.

6. Many government organizations like ICAO (International Civil Aviation Organization) prefer online works now days as working one the online is easy and hassle-free experienceas you need to be careful not to damage the paper ones while the online forms can only be deleted and even if you edit it, you can't change the main columns as they are not in editable form.

7. Instead of renting companies to maintain paper to pen records which cost them annuallyaround $3 billion if they do it on their own, from now on they can save a large amount of $3 billion about earn more profits which increase the company's turnover.

3. REQUIREMENTS

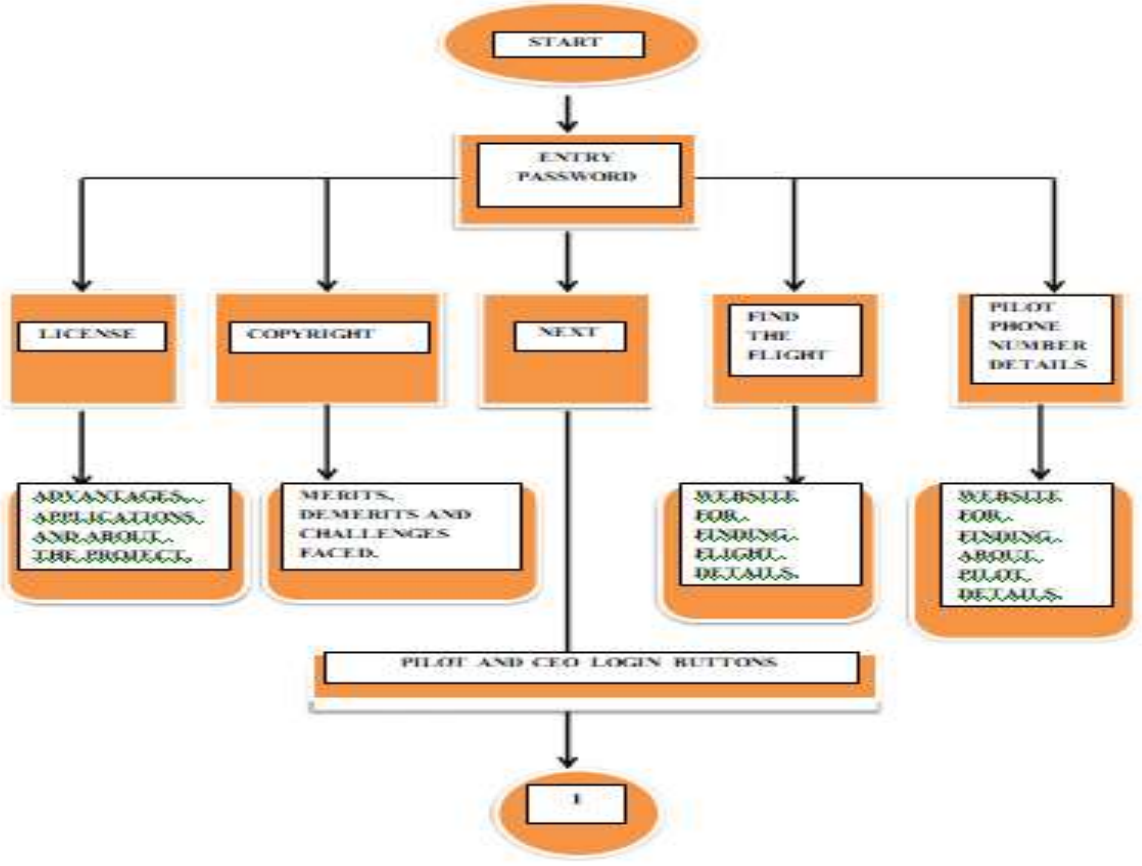
1. Server Infrastructure:
    - At least two high-performance servers for redundancy and failover.
    - Dual or quad-core processors (e.g., Intel Xeon or AMD EPYC).
    - Minimum 16GB of RAM per server.
    - Multiple high-speed network interfaces.
    - Sufficient storage (RAID configuration) for databases and application files.
    - Rack-mountable server chassis for space efficiency.
2. Database Server:
    - Dedicated database server with a powerful CPU.
    - At least 32GB of RAM for efficient data handling.
    - Fast SSD storage or a RAID array for high I/O performance.
    - Backup and redundancy solutions for data integrity.
3. Networking Equipment:
    - Gigabit or 10 Gigabit Ethernet switches with Quality of Service (QoS) capabilities.
    - Redundant network connections for high availability.
    - Network monitoring tools for traffic analysis.
4. Data Backup Systems:
    - Backup servers with similar specifications to the primary servers.
    - Network-attached storage (NAS) or storage area network (SAN) for data backup and recovery.
    - Regular backup schedules and off-site backup storage for disaster recovery.
5. Load Balancers:
    - Load balancers with failover capabilities to distribute traffic evenly.
    - High-speed load balancing algorithms for efficient resource allocation.
6. User Workstations:
    - Standard workstations for administrative staff and passengers with web browsers.
    - Ensure compatibility with modern web technologies and browsers.
7. Security Systems:
    - Hardware firewalls (e.g., Cisco ASA or Palo Alto) with intrusion detection and prevention features.
    - Secure socket layer (SSL) hardware for encrypted communication.
    - Hardware security modules (HSM) for cryptographic key management.

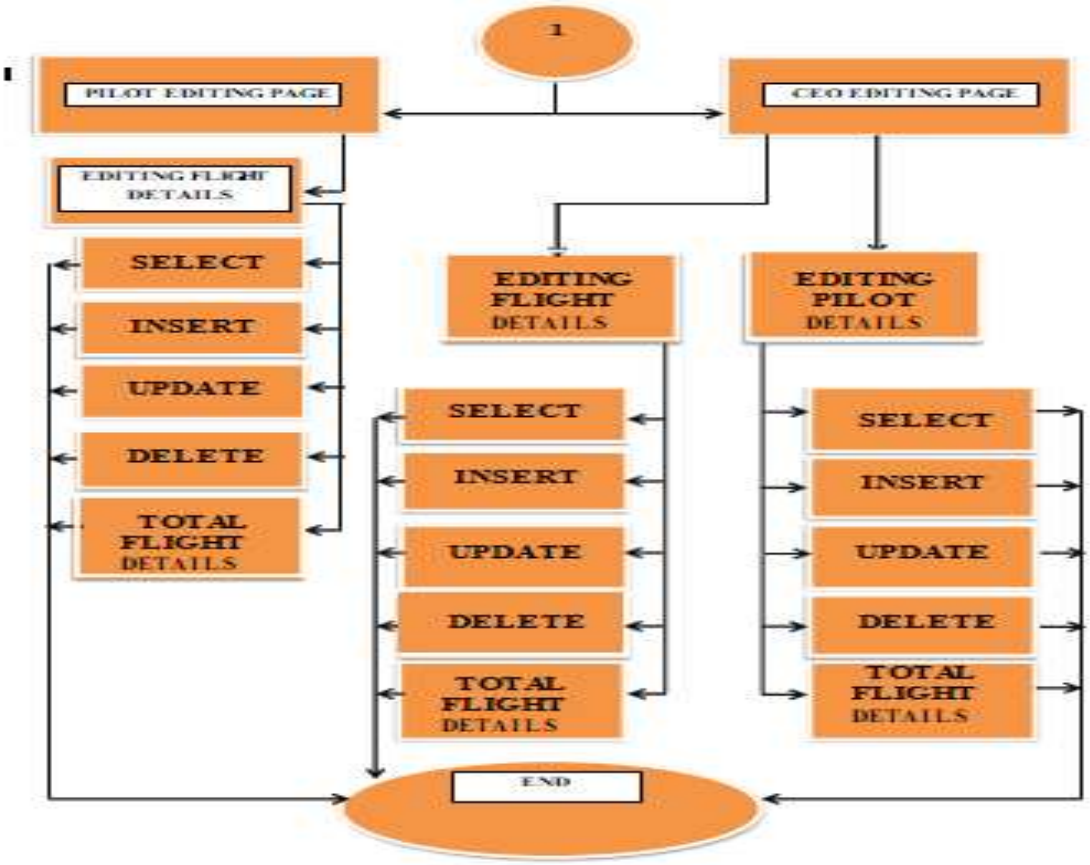
8. Uninterruptible Power Supply (UPS):
    - UPS units with sufficient capacity to ensure uninterrupted system operation during power outages.
    - Generator backup for prolonged power interruptions.
9. Redundancy and Failover Systems:
    - Redundant power supplies and network connections for servers.
    - Hot-swappable components to minimize downtime during hardware failures.
    - Automated failover systems to switch to backup servers in case of primary server failure.
10. Environmental Controls:
    - Air conditioning and ventilation systems for maintaining optimal server room temperatures.
    - Temperature and humidity monitoring equipment.
    - Fire suppression systems to protect server equipment.

4. ARCHITECTURE AND DESIGN

Flowchart:

Flowchart program flow:




1
PILOT EDITING PAGE
CEO EDITING PAGE
EDITING FLIGHT DETAILS
SELECT
INSERT
UPDATE
DELETE
TOTAL FLIGHT DETAILS
EDITING FLIGHT DETAILS
SELECT
INSERT
UPDATE
DELETE
TOTAL FLIGHT DETAILS
EDITING PILOT DETAILS
SELECT
INSERT
UPDATE
DELETE
TOTAL FLIGHT DETAILS
END

5. IMPLEMENTATION

5.1 Code :

app project > ...

```python
#Importing all the necessary modules
import time
import tkinter as tk
from tkinter import *
from tkinter import messagebox
import tkinter.ttk as ttk
import threading
from random import randint
from time import sleep
import webbrowser
def fnd():
    ast=Tk()
    ast.title('ENTRY PAGE')
    def sh():
        e1.delete(0,'end')
    b=Button(ast,text='RESET',command=sh,bd=5,bg='Lemonchiffon')
    b.place(x=350,y=105)
    label3=Label(ast,text="ENTRY PASSWORD",font=['Algerian','30','underline','bold'],bg='gold')
    label3.place(x=80,y=0)
    e1=Entry(ast,text = "",show = '*',font='20')
    e1.place(x=150,y=110)
    print()
    print()

    def bos():
        if e1.get()=='vishal1234':
            ast.destroy()

            '''str3='Start your work sir!!!!'
            ast.destroy()
```

```python
    o=Tk()


    lbl_label=Label(o,text='AIRCRAFT MANAGEMENT SYSTEM',font=['Algerian','28'],bg='lemonchiffon')
    lbl_label.place(x=8,y=20)

    def star():
        o.destroy()
        #***ESTABLISHING CONNECTION***
        #Importing mysql.connector for establishing connection with sql
        import mysql.connector as mc
        con=mc.connect(host="localhost",user="root",passwd="vishal1234",database="vishal")
        cur=con.cursor()
        #****Creating firstWindow***
        wind=Tk()
        def pilot():
            wind.destroy()
            windo=Tk()
            def login():
                if (((name_entry.get())=="Pilot" or (name_entry.get())=="PILOT"or (name_entry.get())=="pilot"or (name_entry.get())
                    messagebox.showinfo("Information","Welcome"+' '''+nam_entry.get()+' '''+"Connected to database successfully")
                    windo.destroy()
                    a=Tk()
        #**INSERT FUNCTION*
                    def insert():
                        b=Tk()
                        def reset():
                            i.delete(0,END)
                            j.delete(0,END)
                            k.delete(0,END)
```

```
            l.delete(0,END)
            m.delete(0,END)
            n.delete(0,END)
            p.delete(0,END)
            q.delete(0,END)
        def back():
            b.destroy()
        def submit():
            sql="insert into project values(%s,%s,%s,%s,%s,%s,%s,%s,%s)"
            values=i.get(),j.get(),k.get(),l.get(),m.get(),n.get(),p.get(),q.get()
            cur.execute(sql,values)
            con.commit()
            messagebox.showinfo("Information","Successfully inserted flight details sir")

        lbl_label=Label(b,text='DATE',font=['Algerian','10'])
        lbl_label.place(x=100,y=50)
        lbl_label=Label(b,text='REGISTRATION_NO',font=['Algerian','10'])
        lbl_label.place(x=100,y=80)
        lbl_label=Label(b,text='PILOT_NAME',font=['Algerian','10'])
        lbl_label.place(x=100,y=110)
        lbl_label=Label(b,text='AIRCRAFT_TYPE',font=['Algerian','10'])
        lbl_label.place(x=100,y=140)
        lbl_label=Label(b,text='AIRCRAFT_IDENT',font=['Algerian','10'])
        lbl_label.place(x=100,y=170)
        lbl_label=Label(b,text='FLIGHT_FROM',font=['Algerian','10'])
        lbl_label.place(x=100,y=200)
        lbl_label=Label(b,text='FLIGHT_TO',font=['Algerian','10'])
        lbl_label.place(x=100,y=230)
        lbl_label=Label(b,text='REMARKS_AND_ENDORSEMENTS',font=['Algerian','10'])
```

```
        lbl_label.place(x=100,y=230)
        lbl_label=Label(b,text='REMARKS_AND_ENDORSEMENTS',font=['Algerian','10'])
        lbl_label.place(x=100,y=260)
        i = Entry(b,text = "",font='16')
        i.place(x=350,y=50)
        j = Entry(b,text = "",font='16')
        j.place(x=350,y=80)
        k = Entry(b,text = "",font='16')
        k.place(x=350,y=110)
        l = Entry(b,text = "",font='16')
        l.place(x=350,y=140)
        m= Entry(b,text = "",font='16')
        m.place(x=350,y=170)
        n = Entry(b,text = "",font='16')
        n.place(x=350,y=200)
        p= Entry(b,text = "",font='16')
        p.place(x=350,y=230)
        q= Entry(b,text = "",font='16')
        q.place(x=350,y=260)
        login_btn=Button(b,text = 'SUBMIT',command=submit,bd=1)
        login_btn.place(x=500,y=435)
        u=Button(b,text = 'BACK',command=back,bd=1)
        u.place(x=200,y=380)
        py=Button(b,text = 'RESET',command=reset,bd=1)
        py.place(x=100,y=380)

        b.title("INSERT NEW FLIGHT DETAILS")
        b.geometry('600x500+300+50')
        b.config(bg="black")
```

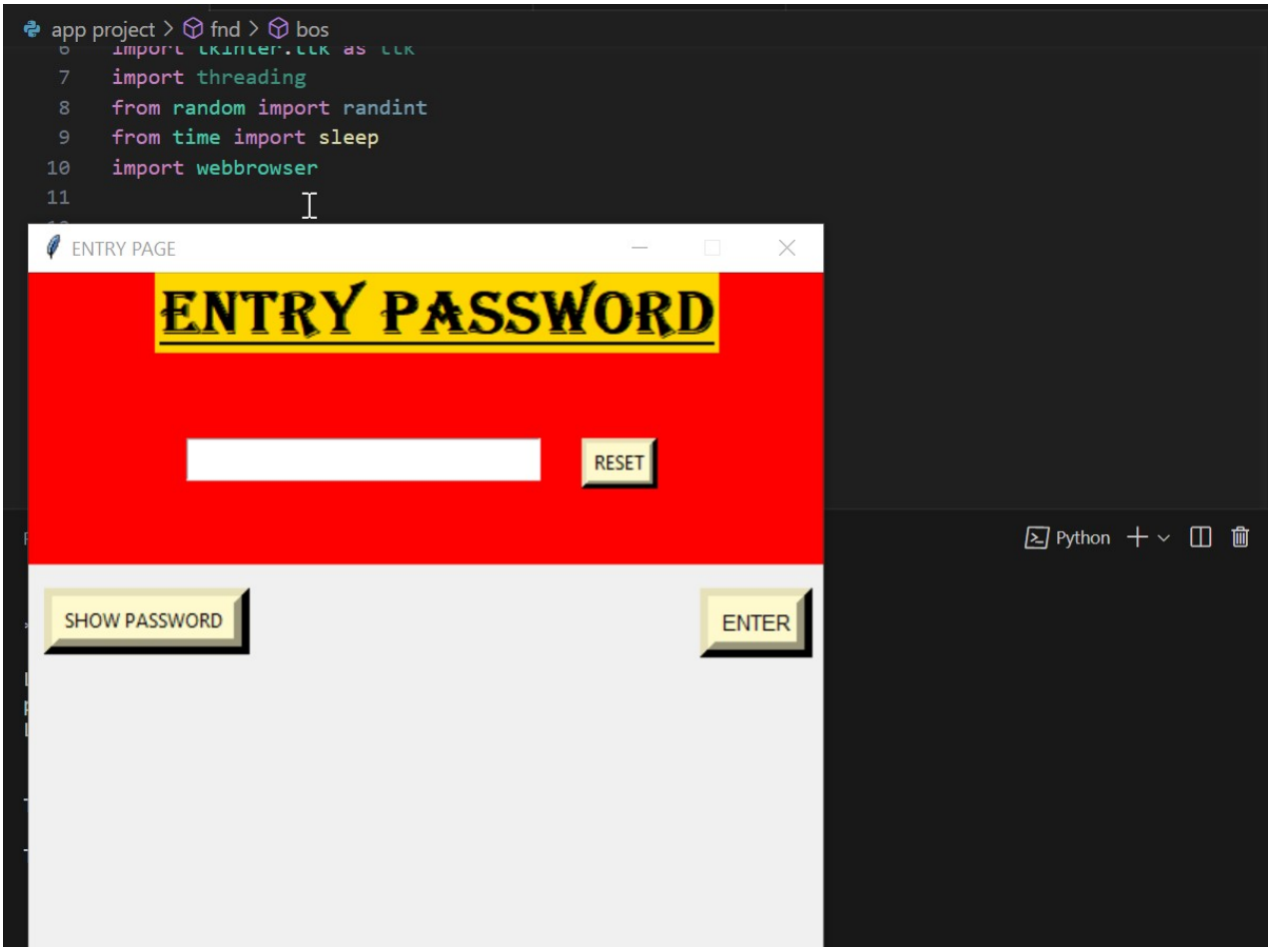
```
                b.mainloop()
        #***UPDATE FUNCTION***
                def update():
                    root=Tk()
                    def back():
                        root.destroy()
                    def fetch():
                        sql="select * from project where Registration_no='%s'"%(e2.get())
                        cur.execute(sql)
                        rows=cur.fetchall()
                        if len(rows)!=0:
                            for row in rows:
                                e1.insert('end',row[0])
                                e3.insert('end',row[2])
                                e4.insert('end',row[3])
                                e5.insert('end',row[4])
                                e6.insert('end',row[5])
                                e7.insert('end',row[6])
                                e8.insert('end',row[7])
                        else:
                            messagebox.showerror(parent=root,title="Error",message="No records found with Billing ID you enter

                    def submit():
                        sql='update project set Date="%s",Pilot_name="%s",Aircraft_type="%s",Aircraft_ident="%s",Flight_from="
                        cur.execute(sql)
                        e1.delete(0,END)
                        e2.delete(0,END)
                        e3.delete(0,END)
                        e4.delete(0,END)
```

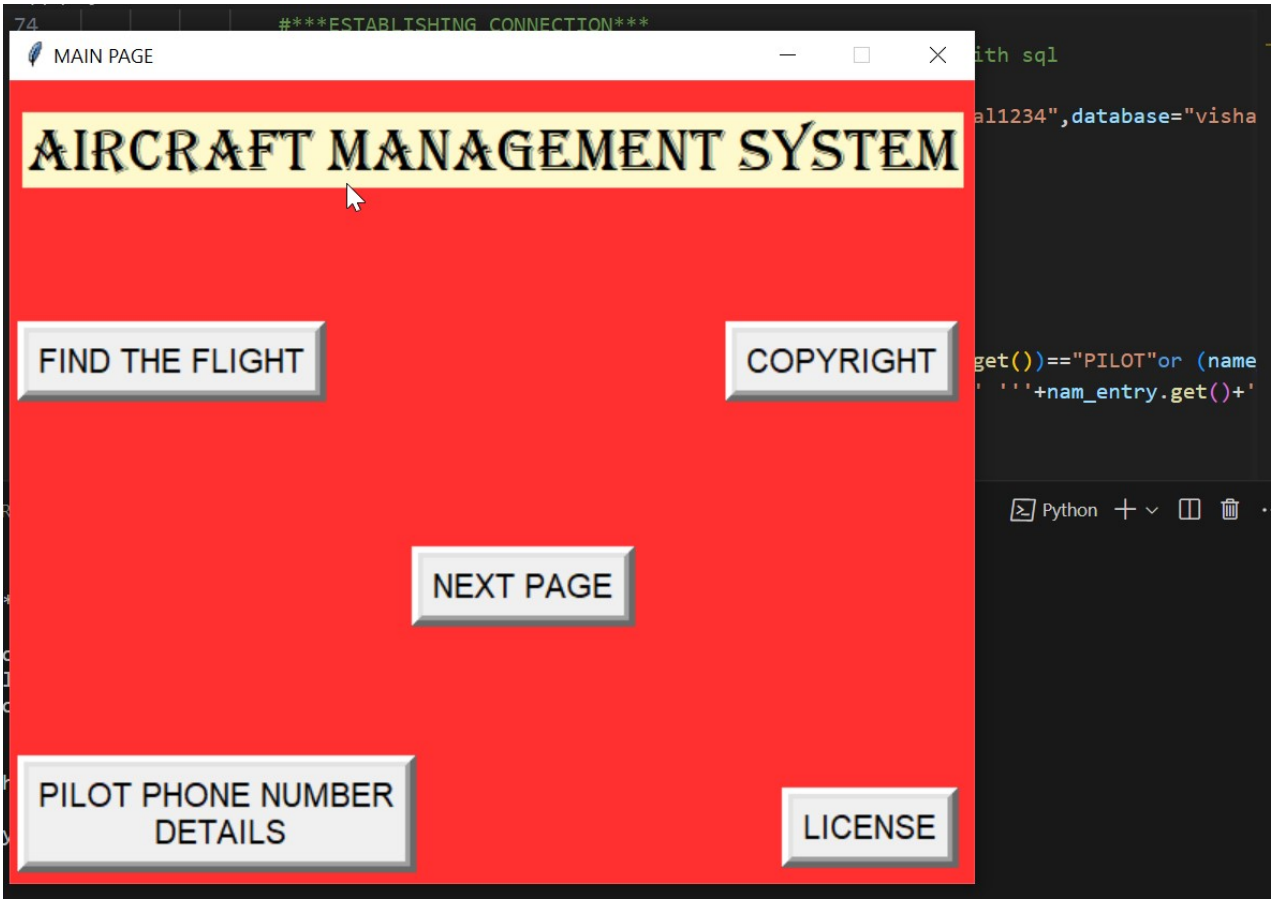
```
                        e2.delete(0,END)
                        e3.delete(0,END)
                        e4.delete(0,END)
                        e5.delete(0,END)
                        e6.delete(0,END)
                        e7.delete(0,END)
                        e8.delete(0,END)
                        messagebox.showinfo(parent=root,title="SUCCESS",message="Record updated successfully")
                        con.commit()

                    label1=Label(root,text="UPDATING FLIGHT DETAILS")
                    label1.place(x=100,y=50)
                    label2=Label(root,text="DATE",font=['Algerian','10'])
                    label2.place(x=100,y=110)
                    label3=Label(root,text="REGISTRATION_NO",font=['Algerian','10'])
                    label3.place(x=100,y=80)
                    label4=Label(root,text="PILOT_NAME",font=['Algerian','10'])
                    label4.place(x=100,y=140)
                    label5=Label(root,text="AIRCRAFT_TYPE",font=['Algerian','10'])
                    label5.place(x=100,y=170)
                    label6=Label(root,text="AIRCRAFT_IDENT",font=['Algerian','10'])
                    label6.place(x=100,y=200)
                    label7=Label(root,text="FLIGHT_FROM",font=['Algerian','10'])
                    label7.place(x=100,y=230)
                    label12=Label(root,text="FLIGHT_TO",font=['Algerian','10'])
                    label12.place(x=100,y=260)
                    label13=Label(root,text="REMARKS AND ENDORSEMENTS",font=['Algerian','10'])
                    label13.place(x=100,y=290)
```

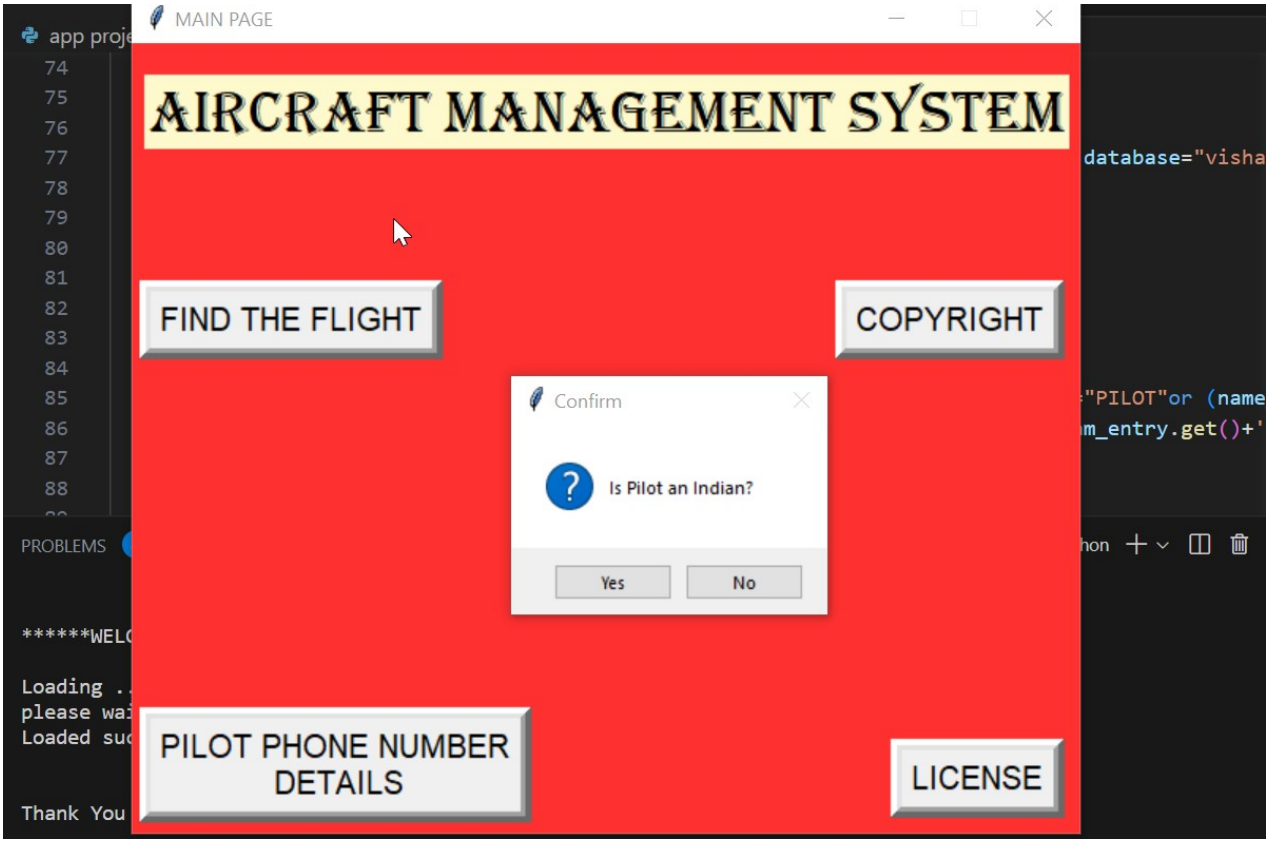
```
            e2.delete(0,END)
            e3.delete(0,END)
            e4.delete(0,END)
            e5.delete(0,END)
            e6.delete(0,END)
            e7.delete(0,END)
            e8.delete(0,END)
            messagebox.showinfo(parent=root,title="SUCCESS",message="Record updated successfully")
            con.commit()

        label1=Label(root,text="UPDATING FLIGHT DETAILS")
        label1.place(x=100,y=50)
        label2=Label(root,text="DATE",font=['Algerian','10'])
        label2.place(x=100,y=110)
        label3=Label(root,text="REGISTRATION_NO",font=['Algerian','10'])
        label3.place(x=100,y=80)
        label4=Label(root,text="PILOT_NAME",font=['Algerian','10'])
        label4.place(x=100,y=140)
        label5=Label(root,text="AIRCRAFT_TYPE",font=['Algerian','10'])
        label5.place(x=100,y=170)
        label6=Label(root,text="AIRCRAFT_IDENT",font=['Algerian','10'])
        label6.place(x=100,y=200)
        label7=Label(root,text="FLIGHT_FROM",font=['Algerian','10'])
        label7.place(x=100,y=230)
        label12=Label(root,text="FLIGHT_TO",font=['Algerian','10'])
        label12.place(x=100,y=260)
        label13=Label(root,text="REMARKS AND ENDORSEMENTS",font=['Algerian','10'])
        label13.place(x=100,y=290)
```

6. RESULTS AND DISCUSSION

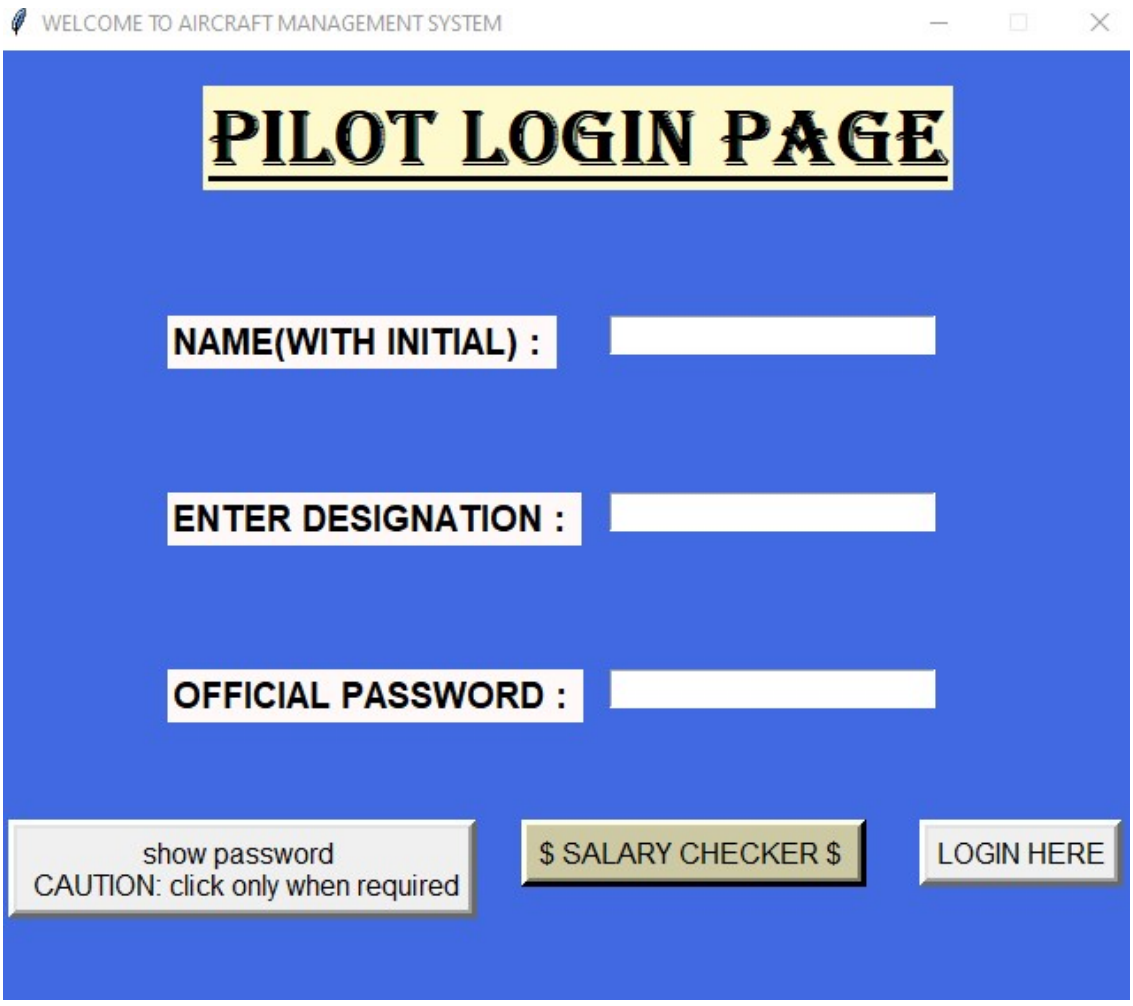
6.1 Results – Snapshots:



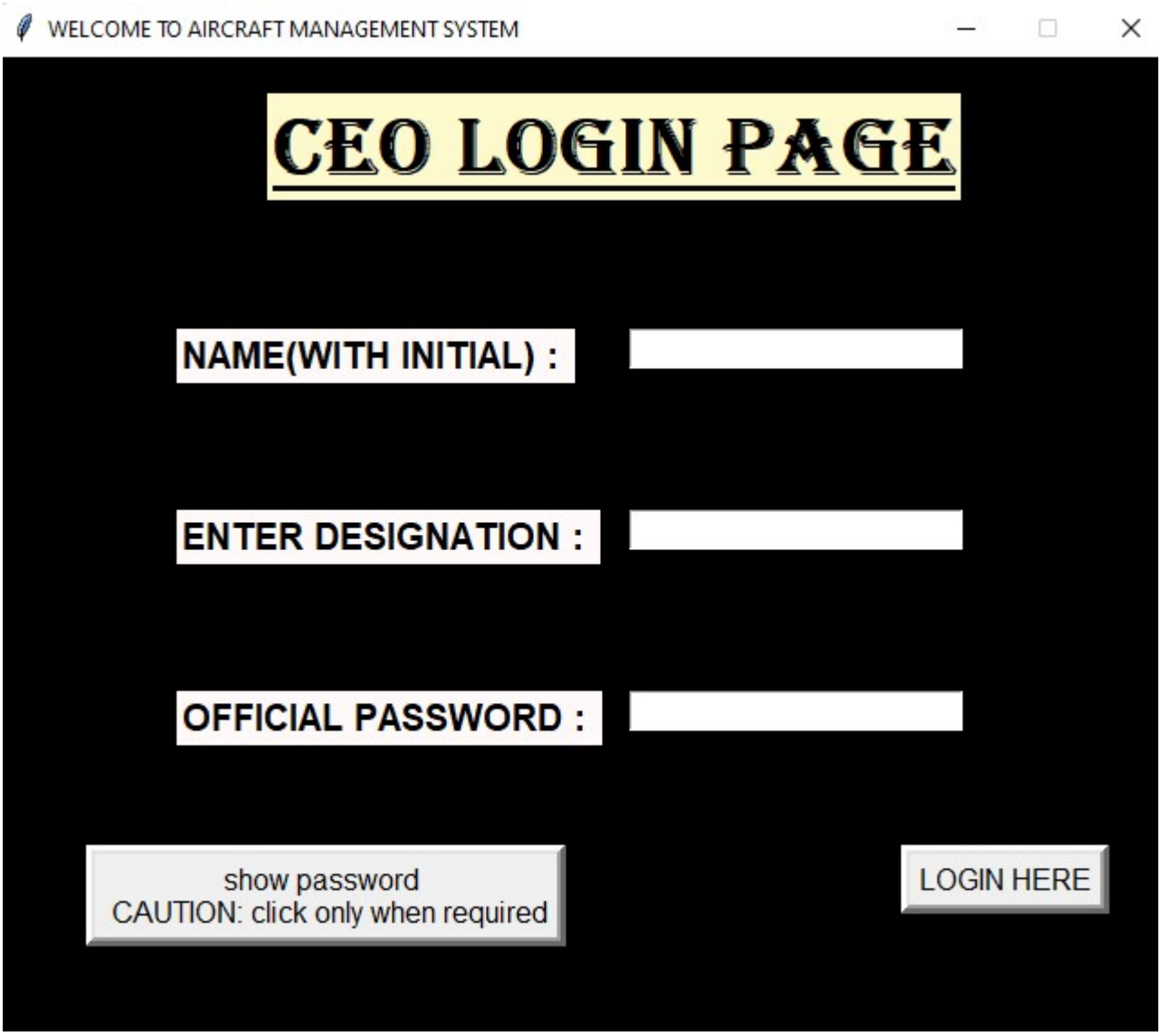

MAIN PAGE
AIRCRAFT MANAGEMENT SYSTEM
FIND THE FLIGHT
COPYRIGHT
NEXT PAGE
PILOT PHONE NUMBER DETAILS
LICENSE


MAIN PAGE
AIRCRAFT MANAGEMENT SYSTEM
FIND THE FLIGHT
COPYRIGHT
Confirm
Is Pilot an Indian?
Yes
No
PILOT PHONE NUMBER DETAILS
LICENSE


MAIN PAGE
AIRCRAFT MANAGEMENT SYSTEM
FIND THE FLIGHT
COPYRIGHT
Confirm
Do you have Internet connection?
Yes
No
PILOT PHONE NUMBER DETAILS
LICENSE
database="visha
"PILOT"or (name
m_entry.get()+'
PROBLEMS
Loading
Thank You


WELCOME TO AIRCRAFT MANAGEMENT SYSTEM
PILOT LOGIN PAGE
NAME(WITH INITIAL) :
ENTER DESIGNATION :
OFFICIAL PASSWORD :
show password
CAUTION: click only when required
$ SALARY CHECKER $
LOGIN HERE


WELCOME TO AIRCRAFT MANAGEMENT SYSTEM
IF YOU ARE PILOT LOGIN HERE
IF YOU ARE CEO LOGIN HERE


WELCOME TO AIRCRAFT MANAGEMENT SYSTEM
CEO LOGIN PAGE
NAME(WITH INITIAL) :
ENTER DESIGNATION :
OFFICIAL PASSWORD :
show password
CAUTION: click only when required
LOGIN HERE

7. CONCLUSION

In conclusion, the Aircraft Management System (AMS) is an essential tool for managing the complex operations of an aviation organization. The project aimed to develop an AMS using Python programming language, which has the potential to automate many tasks and reduce errors associated with manual processes.

The project began with an abstract, introduction, literature review, system architecture, design, and methodology. The methodology section detailed the development process, including requirements gathering, design, coding, testing, and deployment. Several testing methods were identified, including flight simulation testing, maintenance testing, crew management testing, airport operations testing, communication testing, data management testing, and user acceptance testing.

The AMS system's design and architecture aimed to provide a user-friendly interface, modular design, and scalability. The system can handle multiple aircraft and their corresponding maintenance schedules, crew management, and flight operations.

8. REFERENCES

1) Computer Science with python class XI & XII – Author Sumita Arora
2) Core python programming - Author Dr. R. Nageshwara Rao
3) An Introduction to python - Author Guido Van Rossum
4) Computer Science with python c/s 12 - Author Preeti Aror