



REEDS WORLD SCHOOL

Senior Secondary, CBSE

*Affiliated to CBSE, New Delhi, Aff. No.1930514
Globus Garden Main Rd 1, Globus Garden,
Kaikolapalayam, Coimbatore – 641048*



ATTENDANCE REGISTER SYSTEM

PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of

AISSCE

PRACTICAL EXAMINATION

Submitted by

CHRIS JERICSON. R

Reg. No: _____

Under the guidance of

Mr. RATHISH C

M.Sc., M.Phil., B.Ed.

DEPARTMENT OF COMPUTER SCIENCE

MARCH 2024

Certificate



CERTIFICATE

This is to certify that the Project entitled “**ATTENDANCE REGISTER SYSTEM**” submitted in partial fulfillment of the requirements for the award of **AISSCE** practical examination in **COMPUTER SCIENCE** to **REEDS WORLD SCHOOL**, Senior Secondary, affiliated to Central Board of Secondary Education, New Delhi, is a record of bonafide work carried out by **CHRIS JERICSON. R (Reg No:)** under my supervision and guidance, that no part of this project has been submitted for the award of any other examination and the work has not been published in popular journal or magazine.

Signature of the Guide

Signature of the Principal

Viva-Voce conducted on: _____

**Signature of the
External Examiner**

**Signature of the
Internal Examiner**

Declaration



DECLARATION

I hereby declare that the training entitled “**ATTENDANCE REGISTER SYSTEM**” submitted to **REEDS WORLD SCHOOL**, Senior Secondary, affiliated to Central Board of Secondary Education, New Delhi, in partial fulfillment of the requirements for **AISSCE** practical examination is an original work and it has not been previously formed the basis for the award of any examination during the period of my study.

Place: Coimbatore

Signature of the Candidate

CHRIS JERICSON. R

Date:

Reg.No :_____

Acknowledgement



ACKNOWLEDGEMENT

Apart from the efforts of me, the success of any project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in successful completion of this project.

I express my sincere thanks to **Mr. MOHANDOSS, Correspondent, REEDS WORLD SCHOOL** for providing me an infrastructure and moral support while carrying out this project in the school.

I express my deep sense of gratitude to **Mrs. ANURADHA M.Sc., B.Ed., PGDCS Principal, REEDS WORLD SCHOOL** who has been continuously motivating and extending their helping hand to us.

My sincere thanks to **Mr. RATHISH C M.Sc., M.Phil., B.Ed.** master In-charge, a Guide, mentor all the above a friend, who critically reviewed my project and helped in solving every problem, occurred during implementation of the project.

I gratefully acknowledge the contribution of the individuals who contributed to bringing this project up to this level, who continues to look after me despite my flaws.

I express my heartfelt gratitude to my parents for constant encouragement while carrying out this project.

The guidance and support received from all the members who contributed and who are contributing to this project, was vital for the success of the project. I am grateful for their constant support and help.

ABSTRACT

The attendance register is a critical document in schools, universities, and workplaces, as it helps to track the attendance of students or employees. However, manually managing this register can be a tedious and error-prone task, particularly for larger organizations. This project aims to develop an automated attendance register management system that simplifies and streamlines the process of tracking attendance and leave records.

This project will leverage technology to automate the attendance-taking process, allowing teachers, professors, or supervisors to quickly record attendance. This project will store attendance data in a centralized database, allowing administrators to easily access and analyze attendance records, including monthly leave records. This project will also be able to generate attendance and leave reports that can be used for various purposes, such as identifying patterns of absenteeism or tardiness, as well as leave usage. Additionally, this project will have the ability to generate attendance summary reports, which can show daily or monthly attendance records of individual students.

Overall, the attendance register management system will reduce the administrative burden of tracking attendance and leave records while providing more accurate and efficient record-keeping, as well as generating useful attendance reports.

Index



TABLE OF CONTENTS

Contents	Page no.
1. Introduction	01
2. System Specification	02
2.1 Hardware Configuration	
2.2 Software Configuration	
3. Software Description	03
3.1 Python (Front-End)	
3.2 MySQL (Back-End)	
4. System Study	04
4.1 Existing System	
4.2 Proposed System	
4.3 Module Description	
5. System Design	06
5.1 System Flow Diagram	
5.2 Database Design	
5.3 Input Design	
5.4 Output Design	
6. Source Code	09
7. Sample Output	35
8. Conclusion	42
9. Future Enhancement	43
10. Bibliography	44

Introduction



1. INTRODUCTION

Attendance register management system is a computer-based software application that allows users to manage and maintain attendance records of students or any group of individuals. This project automates the process of taking attendance, keeping track of absences, and generating reports.

With an attendance register management system, users can easily create and manage multiple attendance registers for different groups, set up schedules for classes or work shifts, and mark attendance for individuals with just a few clicks. This project also allows users to quickly identify individuals who are absent or have a history of absenteeism.

This project can generate various reports, including attendance summaries, daily or monthly attendance reports, and absence reports. These reports provide valuable insights into attendance trends, which can help identify patterns of absenteeism and improve overall attendance management.

Overall, an attendance register management system is an efficient and reliable solution for tracking attendance and managing attendance records, making it an essential tool for schools, colleges and universities.

2. SYSTEM SPECIFICATIONS

2.1 Hardware Configuration:

System	-	Lenovo
System Name	-	Lenovo-PC
Processor	-	Intel(R) Core(TM) i3-10100 CPU
Clock Speed	-	3.60 GHz
RAM Installed	-	12.00 GB
Memory	-	1024 GB
System Type	-	64-bit OS, x64-based processor

2.2 Software Configuration:

Operating System	-	Windows 10 pro
Front-end	-	Python 3.10
Back-end	-	My SQL 5.5

3. SOFTWARE DESCRIPTION

3.1. Python (Front-End):

Python is regarded as one of the best programming languages for front-end web development because it is simple to learn and has a large number of tools and functionalities. There are many benefits of python front end development and here you can explore some of the best reasons why use python for web development. It is also extremely scalable and capable of achieving a wide range of results. Python can be used for a variety of development tasks, such as creating web apps, games, and more. Python's versatility and agility make it the most popular programming language among organisations worldwide. Google, Facebook, Quora, Amazon, Instagram, NASA and others are among the top Python users. Further this blog will lead into questions like why use python, python front end framework and front end development using python. You will see that Python front-end framework includes different types of libraries to design most attractive graphical user interfaces for the applications.

3.2. MySQL (Back-End):

SQL (Structured Query Language) is a popular programming language used to manage and manipulate relational databases. It is often used as a back-end technology in projects because of its many benefits. SQL is designed to manage large amounts of data efficiently and effectively. It can handle complex queries and is highly scalable, making it ideal for handling large datasets. SQL provides powerful tools for ensuring the integrity of your data, including constraints, indexes, and transactions. This makes it easier to maintain accurate and consistent data, which is critical for many applications. SQL is a user-friendly language that is easy to learn and use. It has a simple syntax and many tools for managing data, making it a popular choice for developers. SQL is supported by many database management systems, including MySQL, Oracle, and Microsoft SQL Server. This makes it easy to switch between systems or migrate data from one system to another. Overall, SQL is an efficient, reliable, and widely used technology that is well-suited for managing large amounts of data. As a result, it is often used as a back-end technology in projects that require efficient and secure data management.

4. SYSTEM STUDY

4.1 Existing System:

Manual System is tedious and has lot of paperwork. It is not much accurate and ambiguity exists in the manual system. No. of registers have to be maintained.

4.1.1 Description of The System:

The Attendance Register System is working manually. The current system is very time consuming and costly, because it involve lot of paper work. To manually handle such a system is very difficult task. But now-a-days because of computerization this job is becoming easier. Following are the reason why the current system should be computerized.

- To increase efficiency with reduced cost.
- To reduce the burden of paper work..
- Easy to supervise the whole database.
- To generate reports easily.

4.1.2 Limitation of Current System:

- **Time consumption:** As the records are to be manually maintained it consumes a lot of time.
- **Paper Work:** Lot of paper work is involved as the records are maintained in the files and registers.
- **Storage Requirements:** As files and registers are used the storage space requirement is increased.
- **Less Reliable:** Use of papers for storing valuable data information is not at all reliable.
- **Accuracy:** As the system is in manual there can be a lot of error. The records might get exchanged between different classes or just be missing. Using a database in that will help to maintain the hive data for the management.

4.2 Proposed System:

The system is developed in Python as a Front-end and back-end MySQL Server Database. The system deals with the most popular interface tool, Python IDLE, as Front End. The system provides well-positioned controlled information to the end user.

The database used, i.e., MySQL Server, which is a fast and well-designed tool for the user to easily manipulate.

The system has better information-over-oriented capabilities for each user's skill while interacting with it. The record retrieval is much quicker than the current system. Therefore, it saves the user time for the next task. The user can interact with the system quickly by inserting, deleting, updating records, etc. Not only does the front end provide faster interaction with records, but also the back end database provides proper interaction with records and provides information/messages if the end user makes an error while working.

4.2.1 Advantages of Proposed System:

- **Easy user interface:** The front end of the system provides easy and pre-defined information for the user to access the system and complete the work faster.
- **Persistent storage:** The system is capable of interacting with the database more quickly. The database also has its own built-in features to store the records.
- **Security:** The system provides security for the records from both the system and the database.
- **More reliable:** The storing of valuable data is managed by the software, making it more reliable than manual systems.
- **More accurate:** The system is automated, so there is no risk of human mistakes.

4.3 Module Description:

i) Attendance:

This module of the program is coded to allow the class teacher to take attendance and store it in the database. This module also shows the overall attendance record.

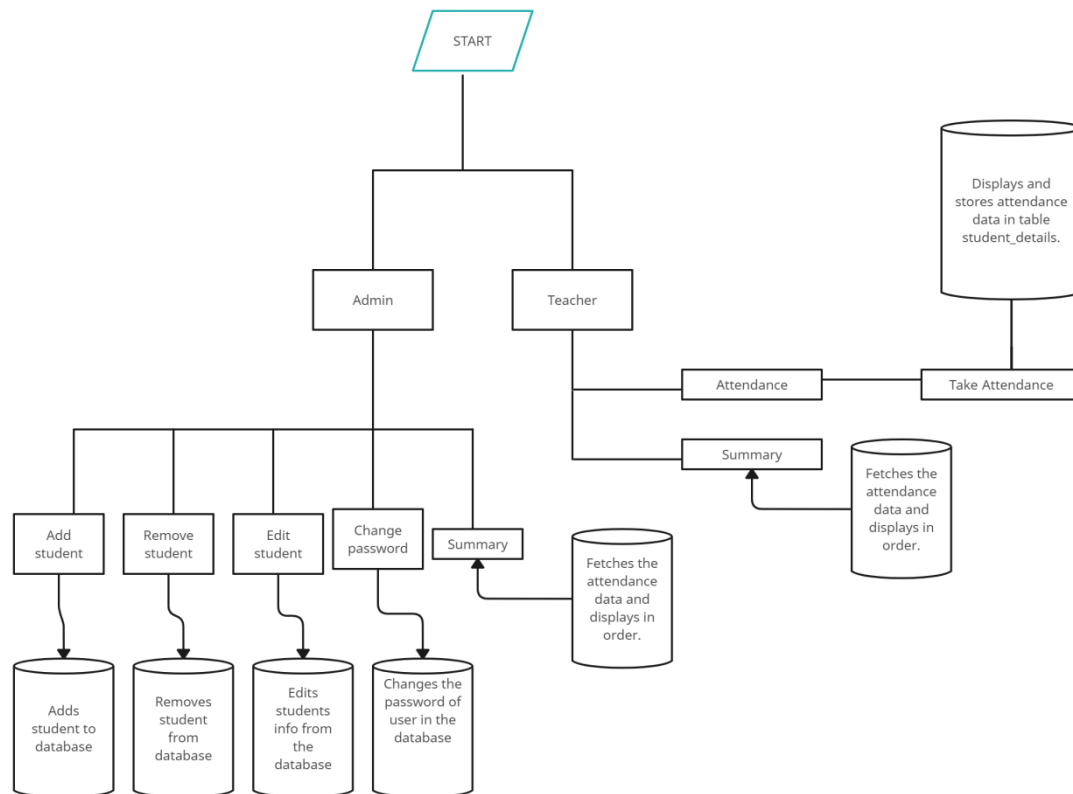
ii) Summary:

This module is coded to allow the class teacher/admin to see the overall summary of the whole class. The one with the most attendance will be at the top and the one with the lowest at the end. This module also allows to view the summary of the class for the particular month.

5. SYSTEM DESIGN

5.1 System Flow Diagram:

System Flow Diagrams are a powerful tool for understanding complex systems. Systems analyses can be anything from business models to biological population models to the effects of social policy and so on. System Diagrams are especially useful in understanding how changes in one factor can affect other factors. Drawing a System Diagram is a great way to begin building a computer model of a system. This technique helps to visualize the structure of a system that you want to model. It shows important factors and relationships and helps you begin to understand the relationships between factors.



5.2 Database Design:

Table name: password

Description: To store the password of the class teacher and the admin

```
mysql> desc password;
```

Field	Type	Null	Key	Default	Extra
designation	varchar(255)	NO		NULL	
name	varchar(255)	NO		NULL	
mobilenno	varchar(20)	NO		NULL	
password	varchar(255)	NO		NULL	

```
4 rows in set (0.00 sec)

mysql> |
```

Table name: student_details

Description: To store the details of the student

```
MySQL 5.5 Command Line Client
mysql> desc student_details;
```

Field	Type	Null	Key	Default	Extra
admission_no	varchar(255)	NO	PRI	NULL	
roll_number	varchar(255)	YES	UNI	NULL	
name	varchar(255)	YES		NULL	
Gender	varchar(10)	YES		NULL	
29-08-2023	varchar(10)	YES		Newbie	
30-08-2023	varchar(10)	YES		Newbie	
11-09-2023	varchar(10)	YES		Newbie	

```
7 rows in set (0.00 sec)

mysql> |
```

5.3 Input Design:

Input design refers to the process of transforming user-created inputs into a computer-readable format. The input data is collected and sorted into a collection of comparable data. Once the relevant data is identified, suitable input media are chosen for processing. The valid data is then stored in the database as operational data. Input design aims to make the entry of input data as simple and error-free as possible. The input screen carefully filters out the invalid data before it becomes operational data at the data entry stage. Input design is one of the most expensive phases of overall system design and requires very close attention. It is also the point of greatest contact between the users and the computer system itself. Input design is prone to errors if the

data entering into the system are incorrect. Processing and output will increase these errors. The goal during input design is

- Collection of data from its source
- Conversion of data into computer acceptable form
- Verification of converted data
- Checking data for its accuracy
- Warning message for wrong entries

In this project the input form has been designed with various tools such as the label box are used to display the username, etc. The buttons and text boxes are used to get the data from the user while making an attempt to login.

The message box has been used to notify the authentication of the admin user. The firm has been designed with navigation menus such as the file, refresh, etc. The dialogue box are also used to represent the details of the consumer and the transformer such as the name of the consumer, address, location, energy level.

5.4 Output Design:

Computer output is the primary and most direct source of information for the user. Effective, clear output design should enhance the system's relationship with the user. Hardcopy from the printer is a major form of output. Output needs are identified during system analysis. The data flow diagram (DFD) is a good starting point for output design. Human factors education issues for design include internal controls to ensure clarity.

Output design defines the output needed and the format it should be produced in. It is important to present the correct information so that the right decisions can be made. The output generated can be divided into three main categories.

- Screen Output
- Output to be stored as file in storage media
- Hard copy of the output

Screen output is basically the generated output shown on screen. Most of the query results are shown on the screen for online information. Providing the generated output to store in the file is a reference. Taking a hard copy of the generated output is a way of providing information to management and what the situation demands.

6. SOURCE CODE

```
# GOD PLEASE HELP ME TO FINISH THIS PROJECT !!!!! :)

import mysql.connector as ms          #ms is used to call mysql.connector

import tkinter

from tkinter import *

import tkinter.messagebox as messagebox

from datetime import *

from tabulate import tabulate

from PIL import Image, ImageTk

database="attendance"    #Change this accordingly

password="02101625"      #Change this according to your mysql password

connector = ms.connect(

    host="localhost",

    user="root",

    password=password,

    database=database

)

cursor = connector.cursor()

cursor.execute("SHOW TABLES LIKE 'password'") # to check weather table exist or not

table_exists = cursor.fetchone() is not None

if table_exists: # not the first time for second time

    cursor.execute("SHOW TABLES LIKE 'student_details'") # to check weather table exist or not
```

table_exists = cursor.fetchone() is not None

if table_exists:

 welcome = Tk() # creating win welcome to show at the begaining

 welcome.configure(bg="black")

 welcome.title("welcome")

 welcome.geometry("800x600")

 welcome.iconbitmap("Reeds.ico")

 screen_width = welcome.winfo_screenwidth()

 screen_height = welcome.winfo_screenheight()

 bg_image = Image.open("voif.jpg")

 bg_image = bg_image.resize((screen_width, screen_height), Image.BICUBIC)

 background_image = ImageTk.PhotoImage(bg_image)

 canvas = Canvas(welcome, width=screen_width, height=screen_height)

 canvas.pack()

 canvas.create_image(0, 0, anchor=NW, image=background_image)

def admin():

 welcome.destroy()

 def geting(): # win asking password window

 global pword # pword user input

 if enterpasswordbox.get() == "":

 messagebox.showwarning('Missing information', 'Please Enter the Password.')

 else:

 pword = enterpasswordbox.get()

```

passwin.destroy()

passwin = Tk() # win to enter password

passwin.configure(bg="black") # bg colour

passwin.title("login")

passwin.geometry("800x600")

passwin.iconbitmap("Reeds.ico")

screen_width = passwin.winfo_screenwidth()

screen_height = passwin.winfo_screenheight()

bg_image = Image.open("voif.jpg")

bg_image = bg_image.resize((screen_width, screen_height), Image.BICUBIC)

background_image = ImageTk.PhotoImage(bg_image)

canvas = Canvas(passwin, width=screen_width, height=screen_height)

canvas.pack()

canvas.create_image(0, 0, anchor=NW, image=background_image)

passwordlabel = Label(passwin, text="Enter password", font=("Arial", 30), fg="white",
bg="black",

                        justify="center", padx=10, pady=10)

passwordlabel.place(relx=0.5, rely=0.4, anchor=CENTER) # password using place

enterpasswordbox = Entry(passwin, show="*", width=30) # Entry box to ask password

enterpasswordbox.place(relx=0.5, rely=0.5, anchor=CENTER)

loginpasswordButton = Button(passwin, text="login", font=("Arial", 15), fg="white",
bg="black",

                        command=geting, width=6, height=1)

```

```

loginpasswordButton.place(relx=0.5, rely=0.6, anchor=CENTER) # login password
button

passwin.mainloop()

cursor.execute("select password from password where designation='admin'")

passadmin=cursor.fetchone()[0]

if pword == "":

    messagebox.showwarning('Missing information', 'Please Enter the Password.')

else:

    if passadmin == pword:

        def restart():

            def summary():

                def yearsumm():

                    summary.destroy()

                    date_column_query = """

                        SELECT COLUMN_NAME

                        FROM INFORMATION_SCHEMA.COLUMNS

                        WHERE TABLE_NAME = 'student_details'

                        AND COLUMN_NAME NOT IN ('admission_no', 'roll_number',

'name', 'Gender')

                    """

                    cursor.execute(date_column_query)

                    date_columns = [column[0] for column in cursor.fetchall()]

# Prepare a dictionary to store student details and attendance counts

```

```

student_details = { }

# Query to fetch student details and attendance for each date column

for date_column in date_columns:

    # Use backticks to reference the date columns

    query = f"SELECT admission_no, roll_number, name, Gender,
`{date_column}` FROM student_details WHERE `{date_column}` IN ('FP', 'AP', 'OD', 'P', 'A')"

    cursor.execute(query)

    attendance_data = cursor.fetchall()

for row in attendance_data:

    admission_no, roll_number, name, gender, attendance_value = row

    if name not in student_details:

        student_details[name] = {'Admission No': admission_no,

                                'Roll No': roll_number, 'Gender': gender,

                                'Present': 0, 'Absent': 0}

    if attendance_value in ['FP', 'AP', 'OD', 'P']:

        student_details[name]['Present'] += 1

    elif attendance_value == 'A':

        student_details[name]['Absent'] += 1

# Sort the student details by the number of days present in descending order

```



```

sorted_student_details = sorted(student_details.items(), key=lambda x:
x[1]['Present'],

reverse=True)

# Prepare data for tabulate

table_data = []

for name, details in sorted_student_details:

    table_data.append(

        [details['Admission No'], details['Roll No'], details['Gender'], name,

        details['Present'], details['Absent']])

# Create a Tkinter window

root = Tk()

root.title("Attendance Summary")

root.geometry("800x600")

root.iconbitmap("Reeds.ico")

# Create a canvas with a vertical scrollbar

canvas = Canvas(root)

canvas.pack(side="left", fill="both", expand=True)

# Create a vertical scrollbar

scrollbar = Scrollbar(root, orient="vertical", command=canvas.yview)

scrollbar.pack(side="right", fill="y")

```

```

canvas.configure(yscrollcommand=scrollbar.set)

# Convert tabulated data to a string with visually formatted table using
"pretty" format

headers = ['Admission No', 'Roll No', 'Gender', 'Name', 'Days Present', 'Days
Absent']

attendance_table = tabulate(table_data, headers=headers, tablefmt='pretty')

# Create a label to display the tabulated attendance summary inside the
canvas

label = Label(canvas, text=attendance_table, justify="left", font=("Courier",
10))

canvas.create_window((0, 0), window=label, anchor="nw")

# Configure canvas scrolling region

canvas.config(scrollregion=canvas.bbox("all"))

# Start the Tkinter main loop

root.mainloop()

def monthsumm():

    def Jan():

        month.destroy()

        monthn = 1

        date_column_query = ""

```

SELECT COLUMN_NAME

```

FROM
INFORMATION_SCHEMA.COLUMNS

WHERE TABLE_NAME = 'student_details'

AND COLUMN_NAME NOT IN
('admission_no', 'roll_number', 'name', 'Gender')

"""

cursor.execute(date_column_query)

date_columns = [column[0] for column in cursor.fetchall()]

# Prepare a dictionary to store student details and attendance counts
student_details = { }

# Query to fetch student details and attendance for each date column
for date_column in date_columns:

    # Use backticks to reference the date columns
    column_parts = date_column.split('-')

    month_part = column_parts[1] # Extract the month part

    query = f"SELECT admission_no, roll_number, name, Gender,
`{date_column}` FROM student_details WHERE `{date_column}` IN ('FP', 'AP', 'OD', 'P', 'A')
AND {monthn} = {month_part}"

    cursor.execute(query)

    attendance_data = cursor.fetchall()

    for row in attendance_data:

        admission_no, roll_number, name, gender, attendance_value = row

```

```

        if name not in student_details:

            student_details[name] = {'Admission No': admission_no,

                                     'Roll No': roll_number, 'Gender': gender,

                                     'Present': 0, 'Absent': 0}

    if attendance_value in ['FP', 'AP', 'OD', 'P']:

        student_details[name]['Present'] += 1

    elif attendance_value == 'A':

        student_details[name]['Absent'] += 1

# Sort the student details by the number of days present in descending
order

sorted_student_details = sorted(student_details.items(),

                                key=lambda x: x[1]['Present'], reverse=True)

# Prepare data for tabulate

table_data = []

for name, details in sorted_student_details:

    table_data.append(

        [details['Admission No'], details['Roll No'], details['Gender'], name,

         details['Present'], details['Absent']])

# Create a Tkinter window

root = Tk()

root.title(f"Attendance Summary for Month {monthn}")

```

```

root.geometry("800x600")

root.iconbitmap("Reeds.ico")

# Create a canvas with a vertical scrollbar

canvas = Canvas(root)

canvas.pack(side="left", fill="both", expand=True)


# Create a vertical scrollbar

scrollbar = Scrollbar(root, orient="vertical", command=canvas.yview)

scrollbar.pack(side="right", fill="y")


canvas.configure(yscrollcommand=scrollbar.set)


# Convert tabulated data to a string with visually formatted table using
"pretty" format

headers = ['Admission No', 'Roll No', 'Gender', 'Name', 'Days Present',
           'Days Absent']

attendance_table = tabulate(table_data, headers=headers, tablefmt='pretty')


# Create a label to display the tabulated attendance summary inside the
canvas

label = Label(canvas, text=attendance_table, justify="left",
              font=("Courier", 10))

canvas.create_window((0, 0), window=label, anchor="nw")

```

```

# Configure canvas scrolling region

canvas.config(scrollregion=canvas.bbox("all"))


# Start the Tkinter main loop

root.mainloop()

def Feb():

    month.destroy()

    monthn = 2

    date_column_query = """

        SELECT COLUMN_NAME

        FROM INFORMATION_SCHEMA.COLUMNS

        WHERE TABLE_NAME = 'student_details'

        AND COLUMN_NAME NOT IN ('admission_no',

'roll_number', 'name', 'Gender')

        """

    cursor.execute(date_column_query)

    date_columns = [column[0] for column in cursor.fetchall()]

    # Prepare a dictionary to store student details and attendance counts

    student_details = { }

    # Query to fetch student details and attendance for each date column

    for date_column in date_columns:

        # Use backticks to reference the date columns

        column_parts = date_column.split('-')

```

```

month_part = column_parts[1] # Extract the month part

query = f'SELECT admission_no, roll_number, name, Gender,
`{date_column}` FROM student_details WHERE `{date_column}` IN ('FP', 'AP', 'OD', 'P', 'A')
AND {monthn} = {month_part}'

cursor.execute(query)

attendance_data = cursor.fetchall()

for row in attendance_data:

    admission_no, roll_number, name, gender, attendance_value = row

    if name not in student_details:

        student_details[name] = {'Admission No': admission_no,

                                'Roll No': roll_number, 'Gender': gender,

                                'Present': 0, 'Absent': 0}

    if attendance_value in ['FP', 'AP', 'OD', 'P']:

        student_details[name]['Present'] += 1

    elif attendance_value == 'A':

        student_details[name]['Absent'] += 1

# Sort the student details by the number of days present in descending
order

sorted_student_details = sorted(student_details.items(),

                                key=lambda x: x[1]['Present'], reverse=True)

```

```

# Prepare data for tabulate

table_data = []

for name, details in sorted_student_details:

    table_data.append(

        [details['Admission No'], details['Roll No'], details['Gender'], name,

        details['Present'], details['Absent']])


# Create a Tkinter window

root = Tk()

root.title(f"Attendance Summary for Month {monthn}")

root.geometry("800x600")

root.iconbitmap("Reeds.ico")

# Create a canvas with a vertical scrollbar

canvas = Canvas(root)

canvas.pack(side="left", fill="both", expand=True)


# Create a vertical scrollbar

scrollbar = Scrollbar(root, orient="vertical", command=canvas.yview)

scrollbar.pack(side="right", fill="y")


date_column_query = ""

```

SELECT

COLUMN_NAME


```

FROM
INFORMATION_SCHEMA.COLUMNS

WHERE

TABLE_NAME = 'student_details'

AND

COLUMN_NAME NOT IN ('admission_no', 'roll_number', 'name', 'Gender')

"""

cursor.execute(date_column_query)

date_columns = [column[0] for column in cursor.fetchall()]

# Prepare a dictionary to store student details and attendance counts

student_details = { }


# Query to fetch student details and attendance for each date column

for date_column in date_columns:

    # Use backticks to reference the date columns

    column_parts = date_column.split('-')

    month_part = column_parts[1] # Extract the month part


    query = f"SELECT admission_no, roll_number, name, Gender,
`{date_column}` FROM student_details WHERE `{date_column}` IN ('FP', 'AP', 'OD', 'P', 'A')
AND {monthn} = {month_part}"

    cursor.execute(query)

    attendance_data = cursor.fetchall()

    for row in attendance_data:

        admission_no, roll_number, name, gender, attendance_value = row

```

```

if name not in student_details:

    student_details[name] = {'Admission No': admission_no,

                             'Roll No': roll_number, 'Gender': gender,

                             'Present': 0, 'Absent': 0}

if attendance_value in ['FP', 'AP', 'OD', 'P']:

    student_details[name]['Present'] += 1

elif attendance_value == 'A':

    student_details[name]['Absent'] += 1

# Sort the student details by the number of days present in descending
order

sorted_student_details = sorted(student_details.items(),

                                key=lambda x: x[1]['Present'], reverse=True)

# Prepare data for tabulate

table_data = []

for name, details in sorted_student_details:

    table_data.append(

        [details['Admission No'], details['Roll No'], details['Gender'], name,

         details['Present'], details['Absent']])

# Create a Tkinter window

root = Tk()

```

```

root.title(f"Attendance Summary for Month {monthn}")

root.geometry("800x600")

root.iconbitmap("Reeds.ico")


# Create a canvas with a vertical scrollbar

canvas = Canvas(root)

canvas.pack(side="left", fill="both", expand=True)


# Create a vertical scrollbar

scrollbar = Scrollbar(root, orient="vertical", command=canvas.yview)

scrollbar.pack(side="right", fill="y")


canvas.configure(yscrollcommand=scrollbar.set)


# Convert tabulated data to a string with visually formatted table using
"pretty" format

headers = ['Admission No', 'Roll No', 'Gender', 'Name', 'Days Present',
           'Days Absent']

attendance_table = tabulate(table_data, headers=headers, tablefmt='pretty')


# Create a label to display the tabulated attendance summary inside the
canvas

label = Label(canvas, text=attendance_table, justify="left",
              font=("Courier", 10))

```

```

        canvas.create_window((0, 0), window=label, anchor="nw")

# Configure canvas scrolling region

canvas.config(scrollregion=canvas.bbox("all"))


# Start the Tkinter main loop

(name, rollno, admissionno, gender))

connector.commit()

infoofstudent.destroy()

saved = Tk() # to create win

saved.configure(bg="black") # bg colour

saved.title("login")

saved.geometry("800x600")

saved.iconbitmap("Reeds.ico")

screen_width = saved.winfo_screenwidth()

screen_height = saved.winfo_screenheight()

bg_image = Image.open("voif.jpg")

bg_image = bg_image.resize((screen_width, screen_height),
Image.BICUBIC)

background_image = ImageTk.PhotoImage(bg_image)

canvas1 = Canvas(saved, width=screen_width, height=screen_height)

canvas1.pack()

canvas1.create_image(0, 0, anchor=NW, image=background_image)

textnew="Student "+str(loop)+" Details Saved"

```

```

        savedLabel=Label(saved, text=(textnew),font=("Arial", 30),fg="white"
                           ,bg="black", justify="center",padx=10, pady=10)

        savedLabel.place(relx=0.5, rely=0.5, anchor=CENTER)

        savedbutton=Button(saved, text="Next Student -->",font=("Arial",
15),fg="white",bg="black",command=justforfun, width=20, height=1)

        savedbutton.place(relx=0.5, rely=0.6, anchor=CENTER)

        saved.mainloop()

cursor.execute("""

CREATE TABLE if not exists student_details (

    admission_no VARCHAR(255) PRIMARY KEY,

    roll_number VARCHAR(255) UNIQUE,

    name VARCHAR(255),

    Gender varchar(10)

)""")

infoofstudent=Tk()

infoofstudent.configure(bg="black") # bg colour

infoofstudent.title("Information ")

infoofstudent.geometry("800x600")

infoofstudent.iconbitmap("Reeds.ico")

screen_width = infoofstudent.winfo_screenwidth()

screen_height = infoofstudent.winfo_screenheight()

bg_image = Image.open("voif.jpg")

bg_image = bg_image.resize((screen_width, screen_height), Image.BICUBIC)

background_image = ImageTk.PhotoImage(bg_image)

```

```

canvas1 = Canvas(infoofstudent, width=screen_width, height=screen_height)

canvas1.pack()

canvas1.create_image(0, 0, anchor=NW, image=background_image)

GI = Label(infoofstudent, text = "Enter the information of the student ", font =
("Arial",30),

            fg = "white", bg = "black", justify = "center", padx = 10, pady = 10)

GI.place(relx=0.5, rely=0.2, anchor=CENTER)    #gi is thee general info


namelabel=Label(infoofstudent, text="Name ",font=("Arial", 15),

                fg="white",bg="black", justify="center",padx=10, pady=10)

namelabel.place(relx=0.3, rely=0.4, anchor=CENTER)

stuname = Entry(infoofstudent, width=25)

stuname.place(relx=0.7, rely=0.4, anchor=CENTER)


rollnolabel=Label(infoofstudent, text="Roll Number ",font=("Arial", 15),

                 fg="white",bg="black", justify="center",padx=10, pady=10)

rollnolabel.place(relx=0.3,rely=0.5,anchor=CENTER)

sturollno = Entry(infoofstudent, width=25)

sturollno.place(relx=0.7, rely=0.5, anchor=CENTER)


genderlabel=Label(infoofstudent, text="Gender ",font=("Arial", 15),

                 fg="white",bg="black", justify="center",padx=10, pady=10)

genderlabel.place(relx=0.3, rely=0.6, anchor=CENTER)

genderButtonM =Button(infoofstudent, text="M",font=("Arial", 10),fg="black",

```

```

        bg="white", command=M, width=5, height=1, relief="raised")

genderButtonM.place(relx=0.65, rely=0.6, anchor=CENTER)

genderlabelor = Label(infoofstudent, text="or", font=("Arial", 10),

        fg="white", bg="black", justify="center", padx=10, pady=10)

genderlabelor.place(relx=0.7, rely=0.6, anchor=CENTER)

genderButtonF = Button(infoofstudent, text="F", font=("Arial", 10), fg="black",

        bg="white", command=F, width=5, height=1, relief="raised")

genderButtonF.place(relx=0.75, rely=0.6, anchor=CENTER)


admissionnolabel=Label(infoofstudent, text="Admission Number ", font=("Arial",
15),

        fg="white", bg="black", justify="center", padx=10, pady=10)

admissionnolabel.place(relx=0.3, rely=0.7, anchor=CENTER)

stuadmissionno=Entry(infoofstudent, width=25)

stuadmissionno.place(relx=0.7, rely=0.7, anchor=CENTER)


Nextstudent=Button(infoofstudent, text="Submit Detail Of The Student --
→", font=("Arial", 15), fg="white",

        bg="black", command=get, width=35, height=1)

Nextstudent.place(relx=0.5, rely=0.8, anchor=CENTER)

infoofstudent.mainloop()

loop += 1

def great():

    allset.destroy()

```

```

def admin():

    welcome.destroy()

    def geting(): # win asking password window

        global pword # pword user input

        if enterpasswordbox.get() == "":

            messagebox.showwarning('Missing information', 'Please Enter the Password.')

        else:

            pword = enterpasswordbox.get()

            passwin.destroy()

    passwin = Tk() # win to enter password

    passwin.configure(bg="black") # bg colour

    passwin.title("login")

    passwin.geometry("800x600")

    passwin.iconbitmap("Reeds.ico")

    screen_width = passwin.winfo_screenwidth()

    screen_height = passwin.winfo_screenheight()

    bg_image = Image.open("voif.jpg")

    bg_image = bg_image.resize((screen_width, screen_height), Image.BICUBIC)

    background_image = ImageTk.PhotoImage(bg_image)

    canvas1 = Canvas(passwin, width=screen_width, height=screen_height)

    canvas1.pack()

    canvas1.create_image(0, 0, anchor=NW, image=background_image)

    passwordlabel = Label(passwin, text="Enter password", font=("Arial", 30), fg="white",
bg="black",

```



```

        justify="center", padx=10, pady=10)

passwordlabel.place(relx=0.5, rely=0.4, anchor=CENTER) # password using place

enterpasswordbox = Entry(passwin, show="*", width=30) # Entry box to ask password

enterpasswordbox.place(relx=0.5, rely=0.5, anchor=CENTER)

loginpasswordButton = Button(passwin, text="login", font=("Arial", 15), fg="white",
bg="black",

        command=geting, width=6, height=1)

loginpasswordButton.place(relx=0.5, rely=0.6, anchor=CENTER) # login password
button

passwin.mainloop()

cursor.execute("select password from password where designation='admin'")

passadmin=cursor.fetchone()[0]

if pword ==":

    messagebox.showwarning('Missing information', 'Please Enter the Password.')

else:

    if passadmin == pword:

        global noofstudents

        global studentsstrength

        global allset

        noofstudents = Tk()

        noofstudents.configure(bg="black")

        noofstudents.title("Strength")

        noofstudents.geometry("800x600")

        noofstudents.iconbitmap("Reeds.ico")

```

```

    Letsget.place(relx=0.5, rely=0.55, anchor=CENTER)

    su.mainloop()

pword=enterpasswordbox.get()

name=enternamebox.get()

mob=entermobbox.get()

if pword == name == mob == "":

    messagebox.showwarning('Missing information', 'Please enter the Required
information.')

else:

    adminpass.destroy()

    confirmwin = Tk() # creating win welcome to show at the begaining

    confirmwin.configure(bg="black")

    confirmwin.title("welcome")

    confirmwin.geometry("800x600")

    confirmwin.iconbitmap("Reeds.ico")

    screen_width = confirmwin.winfo_screenwidth()

    screen_height = confirmwin.winfo_screenheight()

    bg_image = Image.open("voif.jpg")

    bg_image = bg_image.resize((screen_width, screen_height), Image.BICUBIC)

    background_image = ImageTk.PhotoImage(bg_image)

    canvas1 = Canvas(confirmwin, width=screen_width, height=screen_height)

    canvas1.pack()

    canvas1.create_image(0, 0, anchor=NW, image=background_image)

```

```

welcomelabel = Label(confirmwin, text="confirm password and mob no",
font=("Arial", 30), fg="white", bg="black",

justify="center", padx=10, pady=10) # welcome lable

welcomelabel.place(relx=0.5, rely=0.45, anchor=CENTER) # welcome using place

Letsget = Button(confirmwin, text="Confirm -->", font=("Arial", 15), fg="white",
bg="black",

command=ctp,width=10, height=1)

Letsget.place(relx=0.5, rely=0.55, anchor=CENTER)

confirmwin.mainloop()

welcome.destroy()

adminpass = Tk() # creating win welcome to show at the begaining

adminpass.configure(bg="black")

adminpass.title("welcome")

adminpass.geometry("800x600")

adminpass.iconbitmap("Reeds.ico")

screen_width = adminpass.winfo_screenwidth()

screen_height = adminpass.winfo_screenheight()

bg_image = Image.open("voif.jpg")

bg_image = bg_image.resize((screen_width, screen_height), Image.BICUBIC)

background_image = ImageTk.PhotoImage(bg_image)

canvas1 = Canvas(adminpass, width=screen_width, height=screen_height)

canvas1.pack()

canvas1.create_image(0, 0, anchor=NW, image=background_image)

```

```

welcomelabel = Label(adminpass, text="Admin Please Enter Your Details", font=("Arial",
30), fg="white", bg="black",

                        justify="center", padx=10, pady=10) # welcome lable

welcomelabel.place(relx=0.5, rely=0.35, anchor=CENTER) # welcome using place

welcomelabel = Label(adminpass, text="Name ", font=("Arial", 15), fg="white",

                        bg="black",justify="center", padx=10, pady=10) # welcome lable

welcomelabel.place(relx=0.3, rely=0.45, anchor=CENTER) # welcome using place

enternamebox = Entry(adminpass, width=20)

enternamebox.place(relx=0.7, rely=0.45, anchor=CENTER)

welcomelabel = Label(adminpass, text="Mobile Number", font=("Arial", 15), fg="white",

                        bg="black",justify="center", padx=10, pady=10) # welcome lable

welcomelabel.place(relx=0.3, rely=0.55, anchor=CENTER) # welcome using place

entermobbox = Entry(adminpass, width=20)

entermobbox.place(relx=0.7, rely=0.55, anchor=CENTER)

welcomelabel = Label(adminpass, text="Password", font=("Arial", 15), fg="white",

                        bg="black",justify="center", padx=10, pady=10) # welcome lable

welcomelabel.place(relx=0.3, rely=0.65, anchor=CENTER) # welcome using place

enterpasswordbox = Entry(adminpass, width=20)

enterpasswordbox.place(relx=0.7, rely=0.65, anchor=CENTER)

Letsget = Button(adminpass, text="Next -->", font=("Arial", 15), fg="white", bg="black",

                  command=confirm,width=10, height=1)

Letsget.place(relx=0.5, rely=0.75, anchor=CENTER)

adminpass.mainloop()

```

```
welcomelabel = Label(welcome, text="WELCOME !!!!", font=("Arial", 30), fg="white",
bg="black",

                        justify="center", padx=10, pady=10) # welcome lable

welcomelabel.place(relx=0.5, rely=0.45, anchor=CENTER) # welcome using place

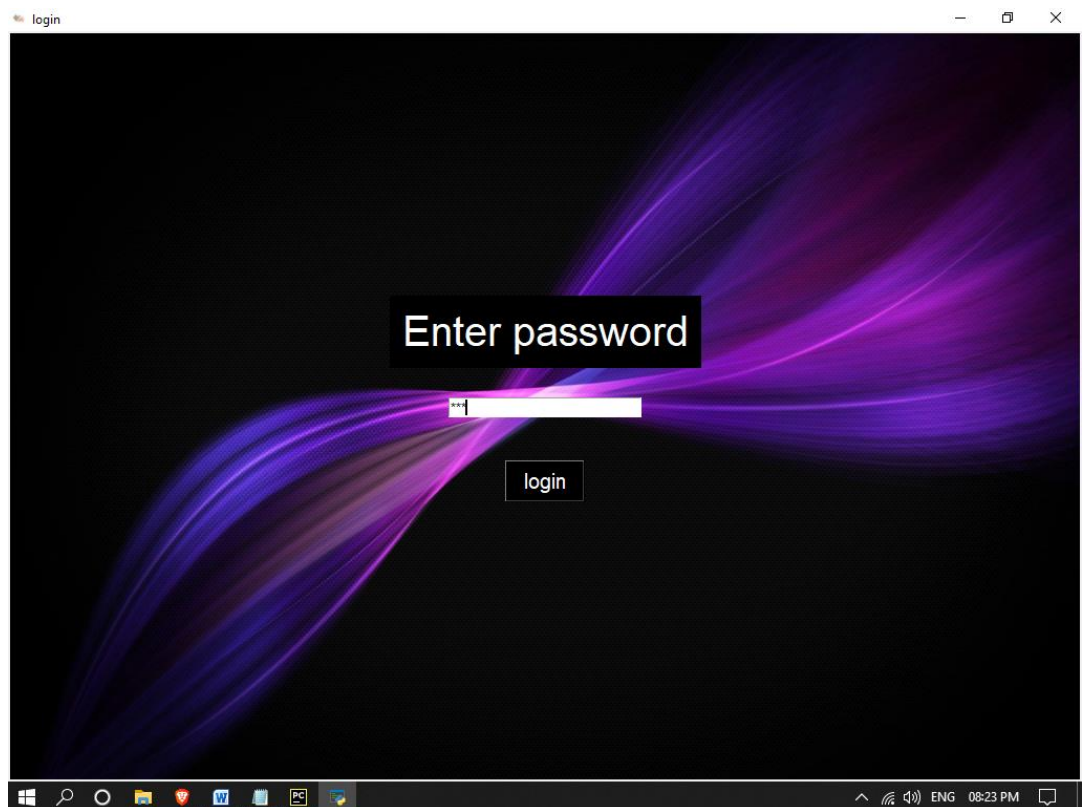
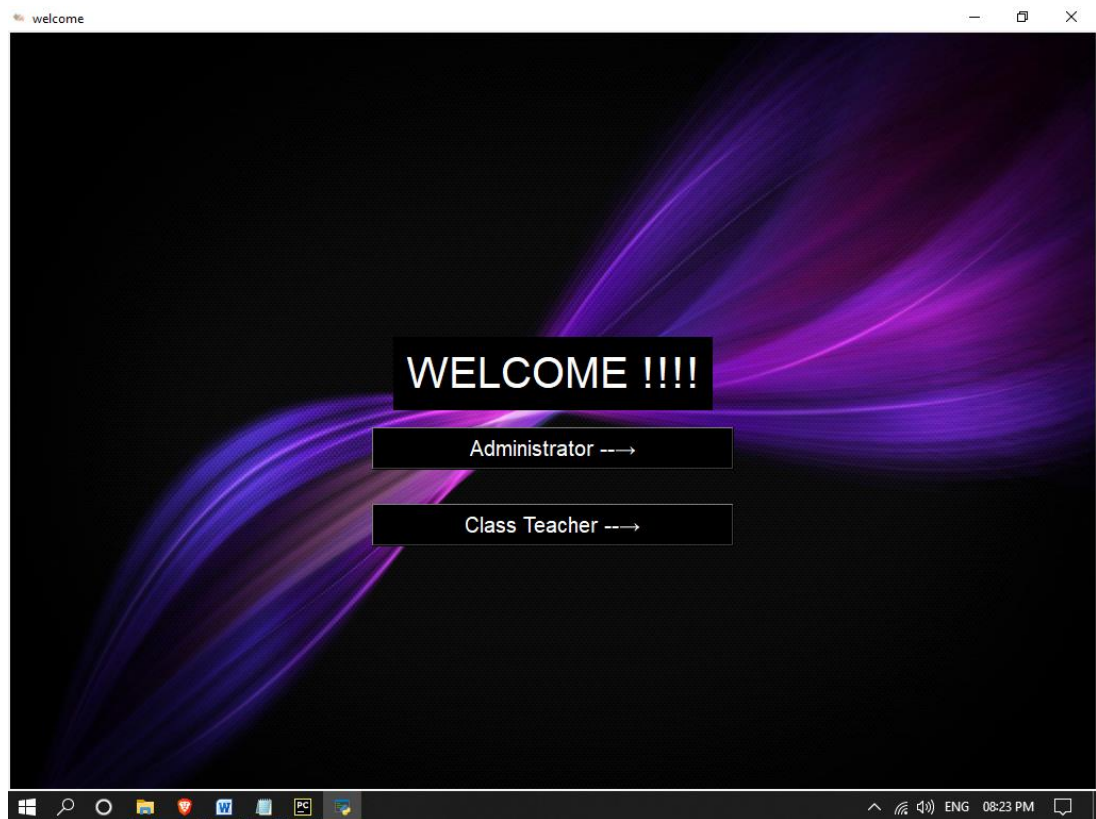
Letsget = Button(welcome, text="Let's Get Started -->", font=("Arial", 15), fg="white",
bg="black", command=letsgo,

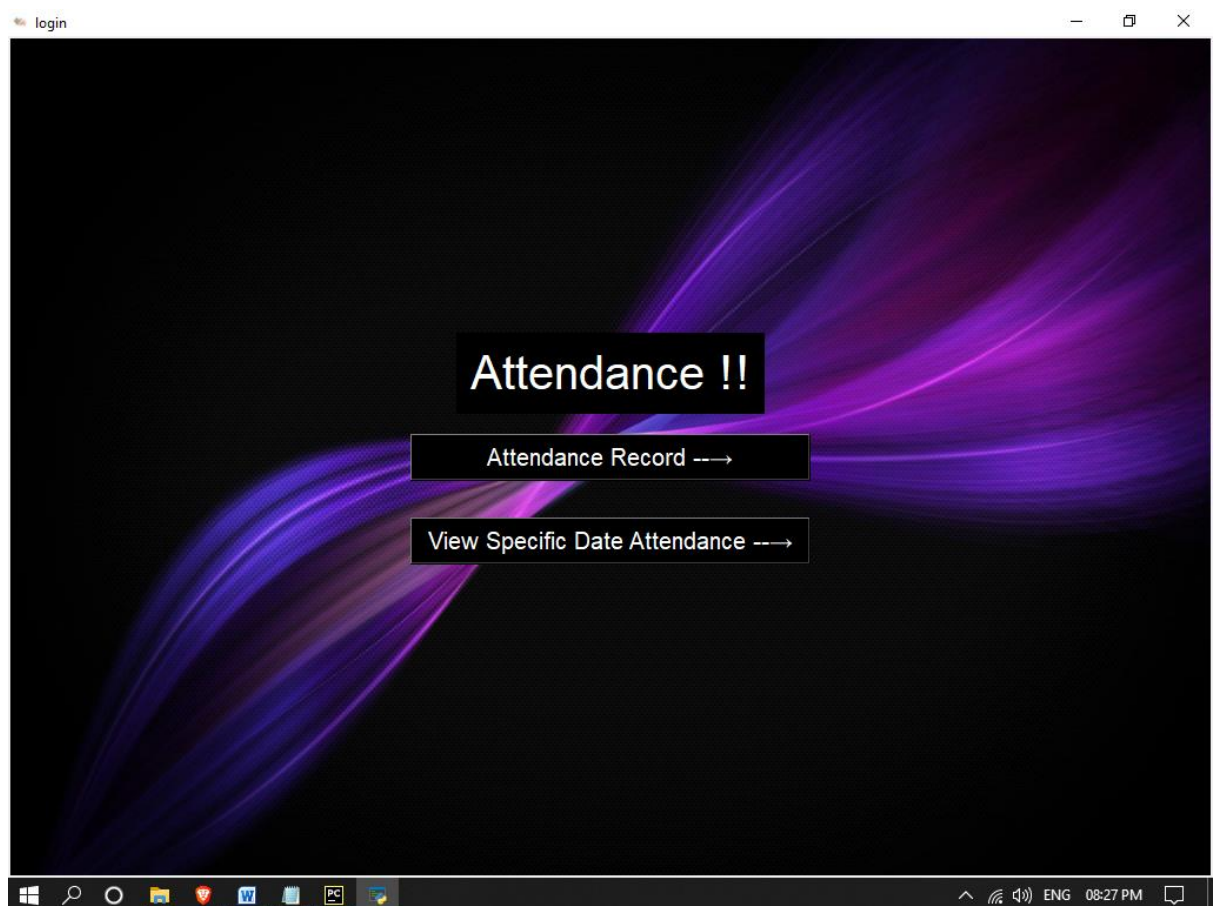
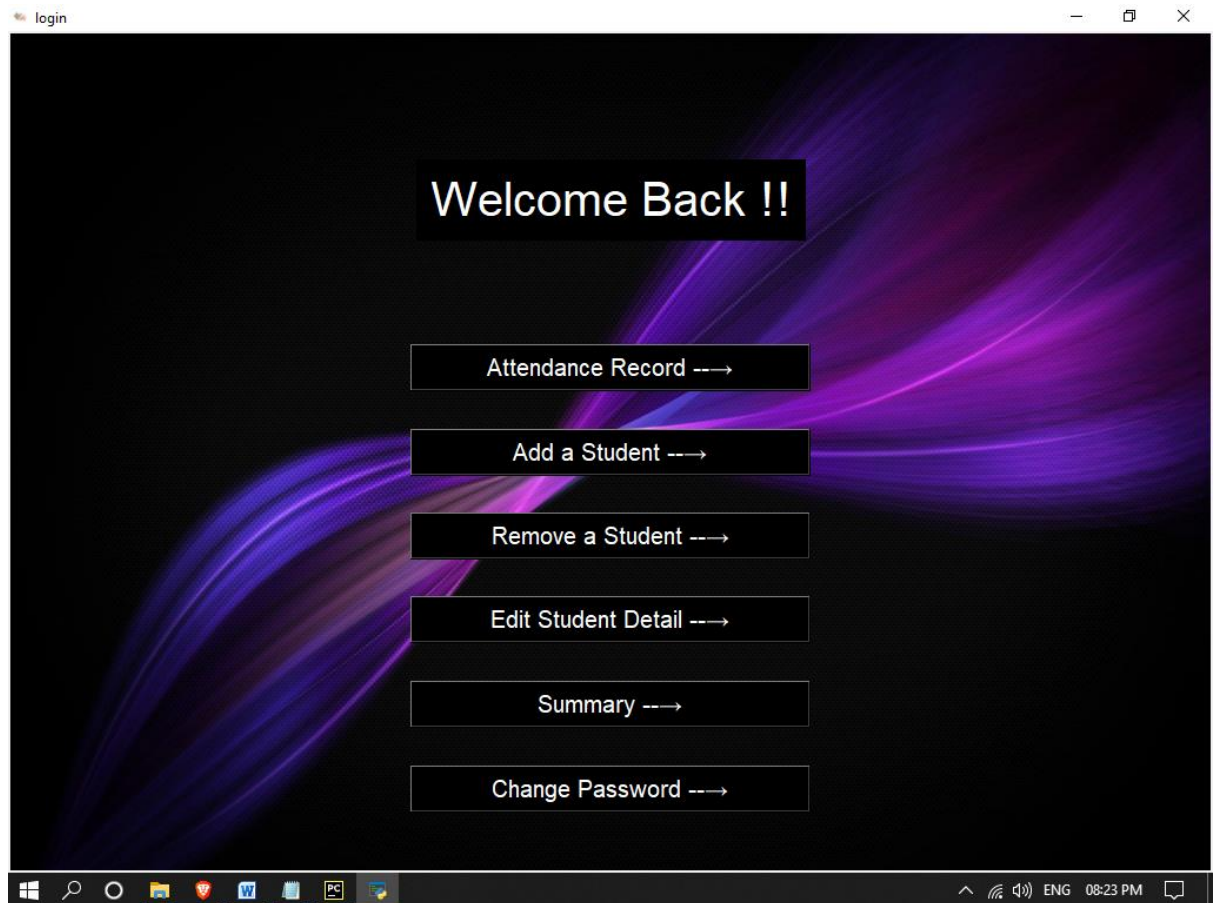
                  width=20, height=1)

Letsget.place(relx=0.5, rely=0.55, anchor=CENTER)

welcome.mainloop()
```

7. SAMPLE OUTPUT





Attendance Report

Name	Roll Number	Admission Number	Gender	11-09-2023
Chris	12A13	RWS17J605	M	P
Lakshith	12A15	RWS17J610	M	A
Survesh	12B08	RWS18M561	M	FP

welcome

Enter the information of the student

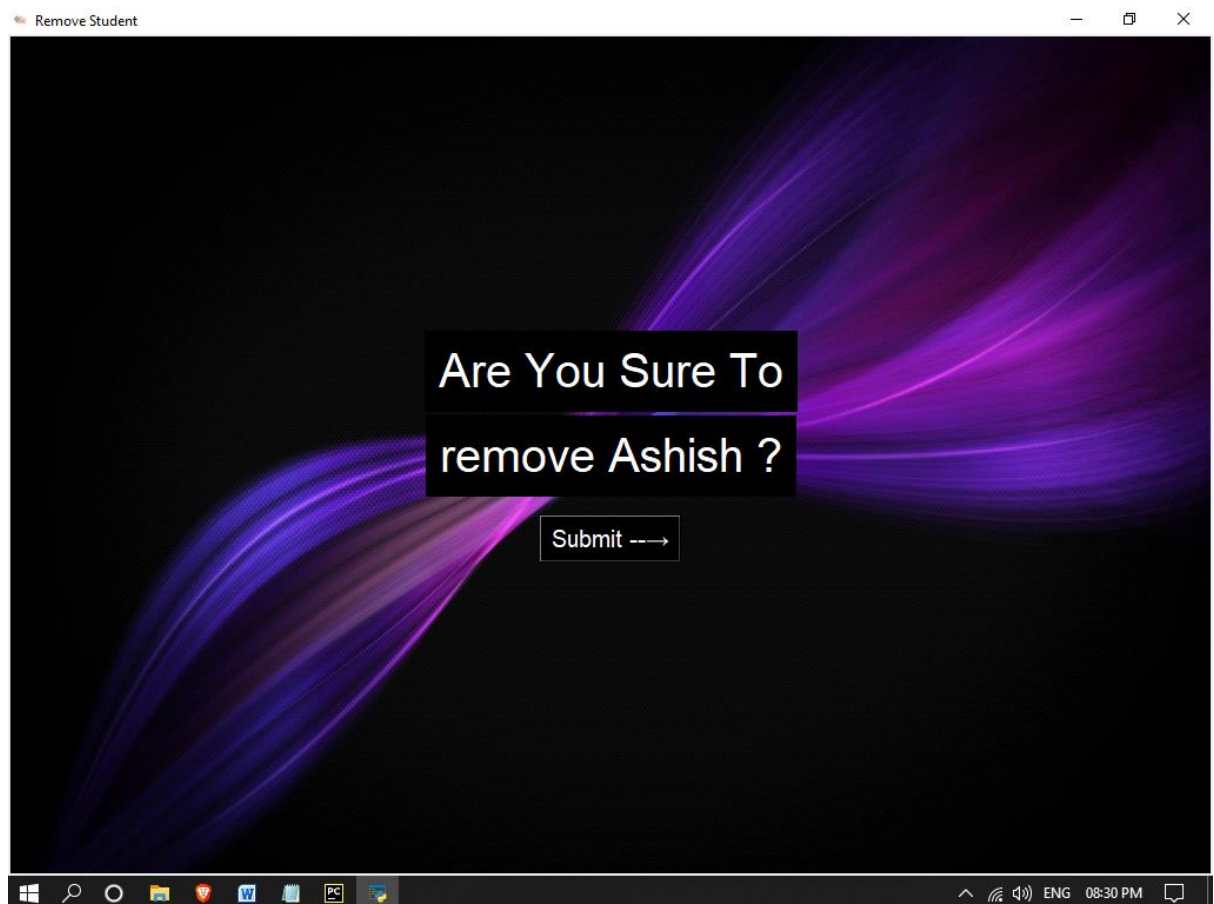
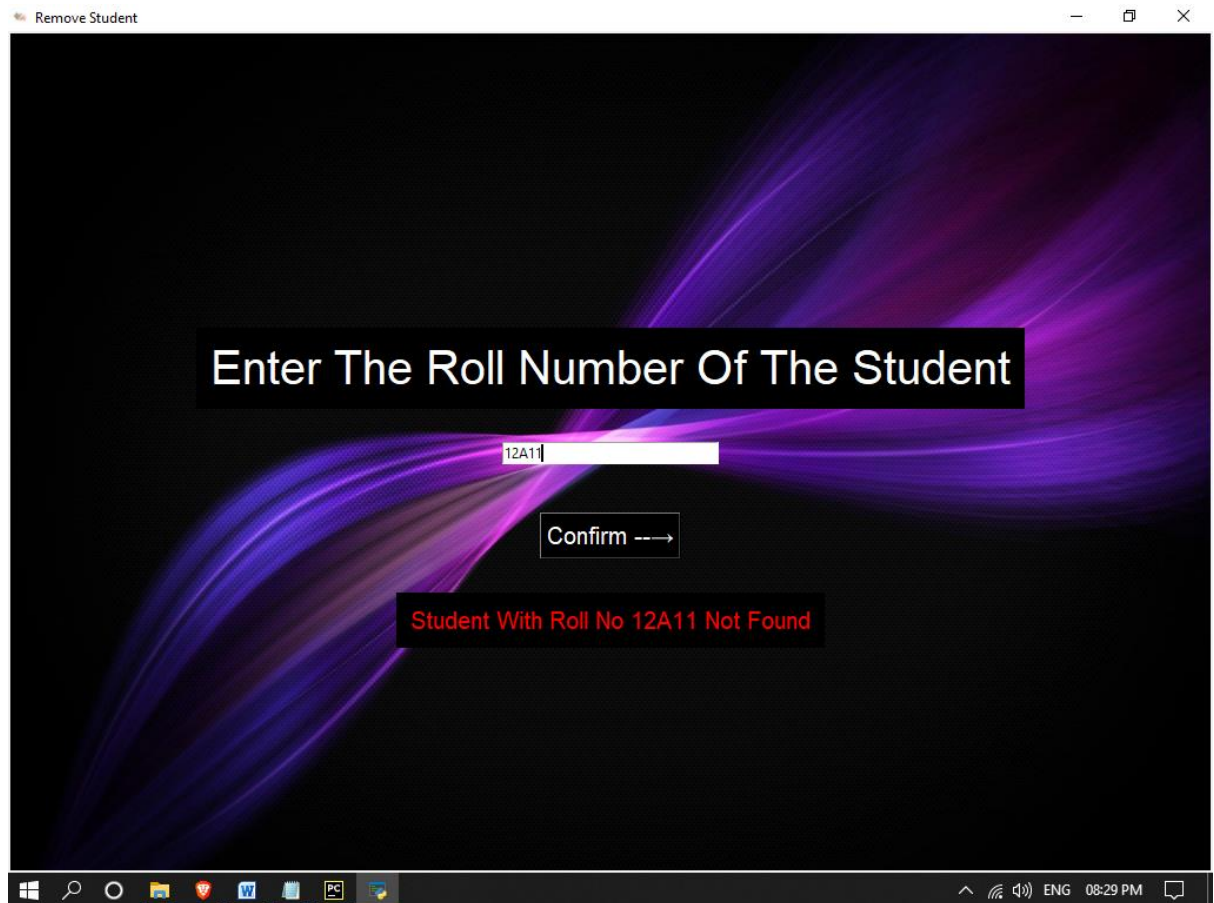
Name

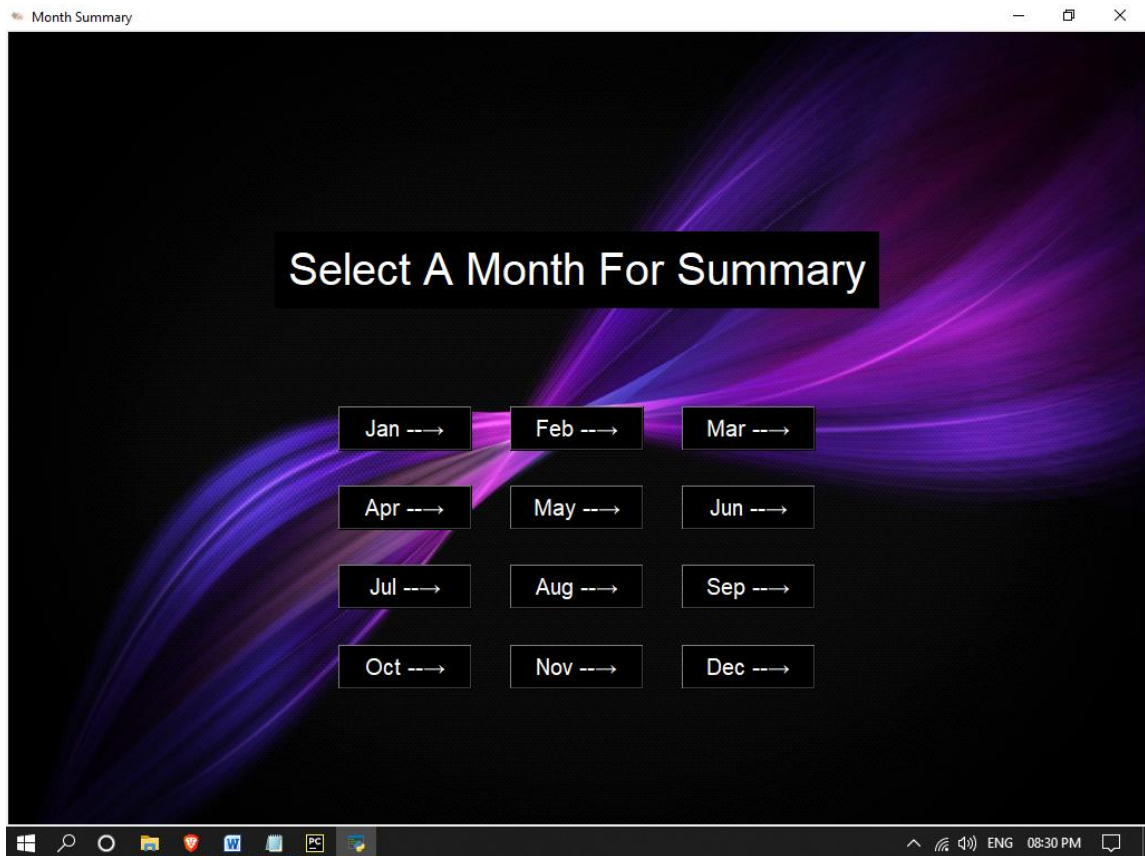
Roll Number

Gender ☐ M or ☐ F **selected M**

Admission Number

[Submit Detail Of The Student -->](#)





Attendance Summary for Month 8

Admission No	Roll No	Gender	Name	Days Present	Days Absent
RWS17J605	12A13	M	Chris	1	1
RWS17J610	12A15	M	Lakshith	1	1
RWS18M561	12B08	M	Survesh	1	0

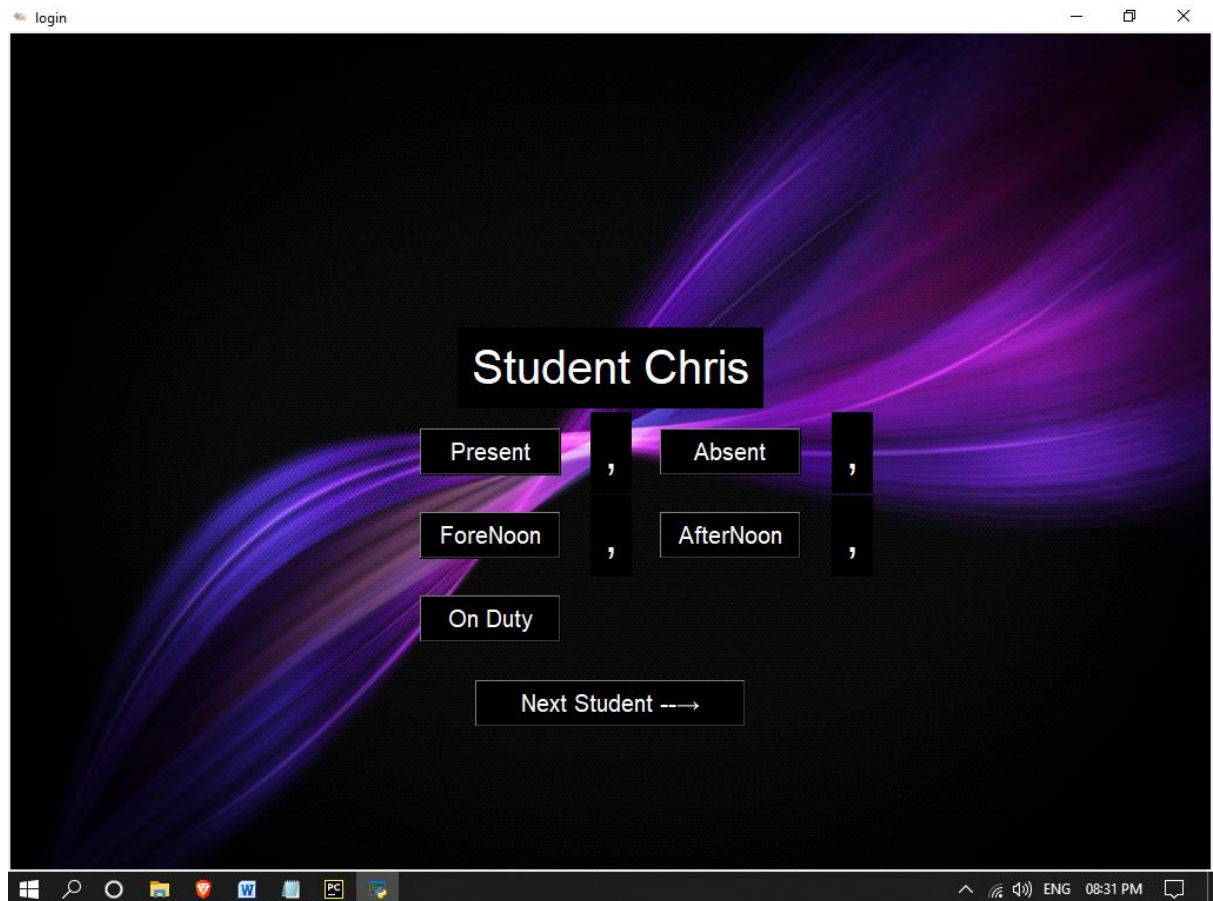
Attendance Summary

Admission No	Roll No	Gender	Name	Days Present	Days Absent
RWS17J605	12A13	M	Chris	2	1
RWS18M561	12B08	M	Survesh	2	0
RWS17J610	12A15	M	Lakshith	1	2

login

Admin! Enter password for Conformation

login



```
mysql> select * from password;
+-----+-----+-----+-----+
| designation | name   | mobileno | password |
+-----+-----+-----+-----+
| admin       | RWS    | 9443701042 | 605      |
| Class Teacher | Rathish | 9791974473 | 604      |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql> select * from student_details;
+-----+-----+-----+-----+-----+-----+-----+-----+
| admission_no | roll_number | name   | Gender | 29-08-2023 | 30-08-2023 | 11-09-2023 | 12-09-2023 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| RWS17J605   | 12A13      | Chris  | M      | P          | A          | P          | P          |
| RWS17J610   | 12A15      | Lakshith | M      | A          | FP         | A          | A          |
| RWS18M561   | 12B08      | Survesh | M      | Newbie     | P          | FP         | FP         |
+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> |
```

8. CONCLUSION

In conclusion, we have discussed how to make the Attendance Register system as simple as possible. It will save time in the future and will make it easier to access and record student's Attendance records.

I have enjoyed working on this novel and difficult project. It has proven to be very useful as it has provided me with practical knowledge not only in Python programming and MySQL-based applications but also in the latest technology that was used in the development of this application. This technology will be in high demand in the future. Therefore, it will provide me with more opportunities and guidance for developing projects on my own.

9. FUTURE ENHANCEMENT

This project aims to improve the old attendance register system, effectively reducing costs, saving time, and enhancing security, as mentioned earlier. Improving the security and integrating servers into the program is the ideal future enhancement plan for this project. Upon enhancement, this project can be rolled out to many other schools and colleges, further enhancing attendance management across educational institutions.

10. BIBLIOGRAPHY

- <https://www.w3schools.com/python/>
- <https://www.tutorialspoint.com/python/index.htm>
- <https://www.programiz.com/python-programming>
- Book reference – Sumita Arora