Login page:

Code-

# Create login window

login\_window = tk.Tk()

login\_window.title("Login")

login\_window.configure(bg="#F4E5FF")

login\_window.geometry('450x200')

# Username entry

Username\_label = tk.Label(login\_window, text="Username:", width=20, font=("Helvetica",15), bg='#E0C3F4')

Username\_label.grid(row=0, column=0) # Align to the east (right)

Username\_entry = tk.Entry(login\_window, width=20, font=("Helvetica",15))

Username\_entry.grid(row=0, column=1)

# Password entry

Password\_label = tk.Label(login\_window, text="Password:", width=20, font=("Helvetica",15), bg='#E0C3F4')

Password\_label.grid(row=2, column=0) # Align to the east (right)

Password\_entry = tk.Entry(login\_window, show="\*", width=20, font=("Helvetica",15))

Password\_entry.grid(row=2, column=1)

# Button to fetch data

fetch\_button = tk.Button(login\_window, text="Login",font=("Helvetica",15), bg='#E0C3F4', command=fetch\_patient\_data)

fetch\_button.grid(row=4, column=0, columnspan=2)

label\_status = tk.Label(login\_window, text="",font=("Helvetica",15),bg="#F4E5FF")

label\_status.grid(row=6, column=0,columnspan=2)

Functionalities used:-

* Label widget- To display text ‘Username’ and ‘Password’.
* Entry widget- To provide the single line text-box to the user to accept a value from the user.
* Grid method- To place the widgets in respective columns and rows after specifying the position.
* Button- To fetch the patient’s data from the table and display it upon calling the fetch\_patient\_data function, passed in the command option.
* .configure method to apply background color
* .geometry method to set the dimensions of the window

Function to fetch Patient data-

Code-

def fetch\_patient\_data():

# Retrieve username and password from entry widgets

Username = Username\_entry.get()

Password = Password\_entry.get()

# Connect to the database

conn = sqlite3.connect('patient\_database.db')

c = conn.cursor()

# Fetch data of the patient with the provided username and password

c.execute("SELECT \* FROM Patient\_information WHERE Username = ? AND Password = ?", (Username, Password))

patient\_data = c.fetchone()

# Close the connection

conn.close()

# Display patient data if found

if patient\_data:

id\_label.config(text="Patient ID: " + str(patient\_data[0]))

name\_label.config(text="Name: " + patient\_data[1])

age\_label.config(text="Age: " + str(patient\_data[4]))

gender\_label.config(text="Gender: " + patient\_data[5])

dob\_label.config(text="Date of Birth: " + patient\_data[6])

bloodGroup\_label.config(text="Blood Group: " + patient\_data[7])

dateVisited\_label.config(text="Date Visited: " + patient\_data[8])

cause\_label.config(text="Cause: " + patient\_data[9])

tablet\_label.config(text="Tablet: " + patient\_data[10])

dose\_label.config(text="Dose: " + str(patient\_data[11]))

NoOfDoses\_label.config(text="No. of Doses: " + str(patient\_data[12]))

ScheduledAppointment\_label.config(text="Appointment scheduled: " + patient\_data[13])

time\_label.config(text="Time: " + patient\_data[14])

login\_window.withdraw() # Hide the login window

appointment\_button = tk.Button(root, text="Take Appointment",bg="#F4E5FF",font=("Helvetica",15), command=take\_appointment)

appointment\_button.grid(row=16, columnspan=2)

else:

label\_status.config(text="Invalid username or password")

login\_window.lift() # Bring the login window to the front

Functionalities used:-

* .get method – to map the input object into a variable which is used further to fetch data
* Created a variable name patient\_data which calls the fetchone() method to fetch one row from the database connected.
* If patient\_data executes-
* Used if statement to display data if patient is found.
* Str- converting the integer datatype to string for displaying
* Using the withdraw() method to hide the login window after successful login
* Shows a button named ‘take appointment’.
* Else statement-
* Prompts a message as “Invalid username or password”
* Lift() method to keep the window in front after the login fails

Function to take appointment:-

Code-

# Function to take appointment

def take\_appointment():

def save\_appointment():

# Retrieve date and time from entry widgets

selected\_date = entry\_date.get()

selected\_time = entry\_time.get()

Username = Username\_entry.get()

# Connect to the database

conn = sqlite3.connect('patient\_database.db')

c = conn.cursor()

# Insert appointment into database

c.execute("UPDATE Patient\_information SET Scheduled\_Appointment = ?, Time = ? WHERE Username = ?", (selected\_date, selected\_time, Username))

conn.commit()

# Close database connection

conn.close()

# Show confirmation message

messagebox.showinfo("Appointment booked successfully!")

# Destroy appointment window

appointment\_window.destroy()

# Create appointment window

appointment\_window = tk.Tk()

appointment\_window.title("Take Appointment")

appointment\_window.geometry('200x200')

appointment\_window.configure(bg="#F4E5FF")

# Date label and entry

label\_date = tk.Label(appointment\_window, text="Date:",font=("Helvetica",15), bg='#F4E5FF')

label\_date.grid(row=0, column=0)

entry\_date = tk.Entry(appointment\_window)

entry\_date.grid(row=0, column=1)

# Time label and entry

label\_time = tk.Label(appointment\_window, text="Time:",font=("Helvetica",15), bg='#F4E5FF')

label\_time.grid(row=1, column=0)

entry\_time = tk.Entry(appointment\_window)

entry\_time.grid(row=1, column=1)

# Save button

save\_button = tk.Button(appointment\_window, text="Save",font=("Helvetica",15), bg='#E0C3F4', command=save\_appointment)

save\_button.grid(row=2, columnspan=2)

Functionalities used-

* Created a function ‘save appointment’ inside the function ‘take appointment’
* Created two variables to store the date and time entered
* Updating the table for only the username entered in the login window using the username variable used in fetch\_patient\_data function.
* Using messagebox window to alert the successful booking of appointment
* .destroy method to hide the appointment window

Take\_appointment function-

* Label widget to display text ‘Date’ and ‘Time’ for taking appointment.
* Entry widget to take one line input for date and time.
* Grid method to align the widgets properly in the window.
* Added a save button which calls the ‘save appointment’ function.

Creating root window-

Code-

# Create main window

root = tk.Tk()

root.title("Patient Data")

root.geometry('350x450')

root.configure(bg="#E0C3F4")

# Labels for displaying patient data

id\_label = tk.Label(root, text="", bg="#E0C3F4",font=("Helvetica",15))

id\_label.grid(row=0, column=0)

name\_label = tk.Label(root, text="", bg="#E0C3F4",font=("Helvetica",15))

name\_label.grid(row=1, column=0)

age\_label = tk.Label(root, text="", bg="#E0C3F4",font=("Helvetica",15))

age\_label.grid(row=2, column=0)

gender\_label = tk.Label(root, text="", bg="#E0C3F4",font=("Helvetica",15))

gender\_label.grid(row=3, column=0)

dob\_label = tk.Label(root, text="", bg="#E0C3F4",font=("Helvetica",15))

dob\_label.grid(row=4, column=0)

bloodGroup\_label = tk.Label(root, text="", bg="#E0C3F4",font=("Helvetica",15))

bloodGroup\_label.grid(row=5, column=0)

dateVisited\_label = tk.Label(root, text="", bg="#E0C3F4",font=("Helvetica",15))

dateVisited\_label.grid(row=6, column=0)

cause\_label = tk.Label(root, text="", bg="#E0C3F4",font=("Helvetica",15))

cause\_label.grid(row=7, column=0)

tablet\_label = tk.Label(root, text="", bg="#E0C3F4",font=("Helvetica",15))

tablet\_label.grid(row=8, column=0)

dose\_label = tk.Label(root, text="", bg="#E0C3F4",font=("Helvetica",15))

dose\_label.grid(row=9, column=0)

NoOfDoses\_label = tk.Label(root, text="", bg="#E0C3F4",font=("Helvetica",15))

NoOfDoses\_label.grid(row=10, column=0)

ScheduledAppointment\_label = tk.Label(root, text="", bg="#E0C3F4",font=("Helvetica",15))

ScheduledAppointment\_label.grid(row=11, column=0)

time\_label = tk.Label(root, text="", bg="#E0C3F4",font=("Helvetica",15))

time\_label.grid(row=12, column=0)

Functionalities used-

* This window contains label widgets to display the information of the pateint such as Patient\_id, Name, Age, Gender, DOB, Blood Group, Date visited, Cause, Tablet name, Doses, If appointment taken or not and Time of appointment.
* Also contains a button which says ‘Take Appointment’ to book an appointment in another window.
* Changing the font style and size, background color by giving the hexadecimal values.

Database Creation:-

Code-

import sqlite3

conn=sqlite3.connect('patient\_database.db')

c = conn.cursor()

conn.execute('''

CREATE TABLE Patient\_information(

Patient\_id INTEGER PRIMARY KEY,

Name TEXT,

Username VARCHAR,

Password INTEGER,

Age INTEGER,

Gender TEXT,

DOB TEXT,

Blood\_group VARCHAR,

Date\_Visited TEXT,

Cause VARCHAR,

Tablet\_Name VARCHAR,

Dose INTEGER,

No\_of\_Doses INTEGER,

Scheduled\_Appointment TEXT,

Time TEXT

);

''')

conn.execute('''

INSERT INTO Patient\_information (Patient\_id,Name,Username,Password,Age,Gender,DOB,Blood\_group,Date\_Visited,Cause,Tablet\_Name,Dose,No\_of\_Doses,Scheduled\_Appointment,Time) VALUES

(111,'Anannya Deshmukh','Ana13',1331,34,'Female','1990-02-13','A','2024-02-21','Cold and Fever','DOLO 650',2,10,'No','No'),

(112,'Poonam Joshi','Poo09',1111,29,'Female','1994-7-23','O+','2024-02-24','Diarrhea','Loperamide',2,10,'2024-02-28','17:15:00'),

(113,'Kunal Pandey','kunal@19',2123,24,'Male','2000-1-19','A','2024-3-1','Migraine','Flunarizine',2,30,'2024-3-16','16:30:00'),

(114,'Anisha Chalke','Anuchalke',1234,39,'Female','1984-5-15','B+','2024-2-9','Hypertension','Olmesar CH40',1,30,'No','No'),

(115,'Vedant Patel','ved10',5678,11,'Male','2013-2-13','A','2024-3-10','Ear Infection','Ciprofloxacin(Ciplox) ear drops',3,15,'No','No');

''')

conn.commit()

Screenshots of the project-







