

About project and Software used

1. This is a basic level hospital management program. It is developed using GUI short for Graphical User Interface. In our program we have used standard Python interface i.e. Tkinter. All the data is stored in MySql DBMS in database named hospital.
2. It can be accessed by either management workers or receptionist or doctors (each doctor have different access i.e. they have unique username and password)
3. The following program involve three individuals in related steps:
 - (i) Installation of the database hospital and creates tables.
 - (ii) Using the program to store or extract information to / from the database created above.
 - (iii) Uninstalling program; this will erase all the data from your DBMS.

4. Our program do maintain few loopholes and there are scope of improvements but we as a team have tried our best to avoid and fix any possible error.

5. This program can be very effective at basic level:

- (i) **Safe:** person who does not have access to MySql or the username or password cannot login.
- (ii) **Easy to understand:** Python and MySql is used which are easy to read and interpret.
- (iii) **Minimizing user's work:** you don't have to create database every time and it will show information boxes(i.e. messagebox) whenever there is an error done by the user.



Codes

Installation

Hospital management system

Uninstallation

Installation Program(Codes):-

```
installation hospital.py - /Users/mac/Downloads/HMS Apoorva, Aayushi and Anubhav 12th Science/installation hospital.py (3.8.5)
__author__ = "APOORVA JADHAV, AAYUSHI KAPOOR & ANUBHAV UTKARSH"
__copyright__ = "Copyright (C) 2021 AAA"
__license__ = "Public Domain"
__version__ = "1.0"

import mysql.connector as mc
from mysql.connector import Error
import tkinter as tk
import tkinter.messagebox as tmb
from tkinter import *

def create1():
    global e, w2d, E
    p = E.get()

    try:
        conn_ob=mc.connect(host="localhost",user="root",password = p)

        if (conn_ob.is_connected()):
            q1="create database if not exists hospital;"
            query_ob=conn_ob.cursor()
            query_ob.execute(q1)
            conn_ob.commit()

            q2="use hospital;"
            query_ob=conn_ob.cursor()
            query_ob.execute(q2)
            conn_ob.commit()

            q3=("create table if not exists patient"+
                 "(pid int primary key not null, pname varchar(30) not null, gender char(1) not null, address varchar(50) not null, blood_group varchar(5) not null,"+
                 "reg_pat_date datetime not null, check(gender in('M','F','O')),check (blood_group in('A+ve','A-ve','B+ve','B-ve','AB+ve','AB-ve','O+ve','O-ve')));")
            query_ob=conn_ob.cursor()
            query_ob.execute(q3)
            conn_ob.commit()

            q4=("create table if not exists doctor"+
                 "(did int primary key not null, dname varchar(30) not null, speciality varchar(50) not null, gender char(1) not null, " +
                 "check(gender in ('M','F','O')));")
            query_ob=conn_ob.cursor()
            query_ob.execute(q4)
            conn_ob.commit()

            q5=("create table if not exists d_patient"+
                 "(pid int references patient(pid), did int references doctor(did), disease varchar(30) not null, iop char(1) not null,"+
                 "status char(1) not null, check (iop in('I','O')));")
            query_ob=conn_ob.cursor()
            query_ob.execute(q5)
```

```
query_ob=conn_ob.cursor()
query_ob.execute(q5)
conn_ob.commit()

q6="create table if not exists login(username char(20) primary key, password char(20), role int(1), id int, check( role in (1,2,3)));"
query_ob=conn_ob.cursor()
query_ob.execute(q6)
conn_ob.commit()

q7="insert into login (username, password, role) values('u1','p1',1),('u2','p2',2);"
query_ob=conn_ob.cursor()
query_ob.execute(q7)
conn_ob.commit()

q8="insert into login values ('u3','p3',3, 1);"
query_ob=conn_ob.cursor()
query_ob.execute(q8)
conn_ob.commit()

q9 = "insert into doctor values(1, 'Test doctor', 'Test Speciality', '0');"
query_ob=conn_ob.cursor()
query_ob.execute(q9)
conn_ob.commit()

tmb.showinfo("Installing...","INSTALLATION COMPLETED !!!")
window1.destroy()

window2 = tk.Tk()

window2.title("3 sample users for testing")
window2.geometry("1000x600")

L1 = tk.Label(window2, text = "3 sample users for testing", font = ("Arial",20)).place(x=0, y=0)
L2 = tk.Label(window2, text = "These are as follows:-", font = ("Arial",20)).place(x=0, y=25)
L3 = tk.Label(window2, text = "1. Username_1 = u1,          Password_1 = p1", font = ("Arial",20)).place(x=0, y=50)
L4 = tk.Label(window2, text = "2. Username_2 = u2,          Password_2 = p2", font = ("Arial",20)).place(x=0, y=75)
L5 = tk.Label(window2, text = "3. Username_3 = u3,          Password_3 = p3      id(did) = 1", font = ("Arial",20)).place(x=0, y=100)

B1 = tk.Button(window2, text = "DONE", font=("ComicSansMS", 20), fg="red", command = window2.destroy).pack(side = 'right')

window2.resizable(False, False)
window2.mainloop()

except Error as e:
    tk.messagebox.showinfo("Error while connecting to MySQL",e)
```

```
installation hospital.py - /Users/mac/Downloads/HMS Apoorva, Aayushi and Anubhav 12th Science/installation hospital.py (3.8.5)

window2.mainloop()

except Error as e:
    tk.messagebox.showinfo("Error while connecting to MySQL",e)

def back1():
    window1.destroy()
    w()

def create():
    global E, window1
    window1 = tk.Tk()

    window1.title("PASSWORD ( MySql )")
    window1.geometry("1000x600")

    l1 = tk.Label(window1, text = "Enter your MySQL password: ", font = ("Arial", 40), bg = "pink", fg = "blue").pack(fill='x')
    E = tk.Entry(window1, width = 50, bg="black", fg="white", show="*")
    E.pack(pady=10)
    l2 = tk.Label(window1, text = "Click next to continue.", font = ("Arial", 20)).pack(side = 'left')
    B1 = tk.Button(window1, text = "Next", font=("ComicSansMS", 20), fg="green", command = create1).place(x=940,y=500)

    B2 = tk.Button(window1, text = "Back", font=("ComicSansMS", 20), fg="red", command = back1).place(x=10, y=500)

    window1.resizable(False, False)
    window1.mainloop()

def next1():
    window.destroy()
    create()

def w():
    global window
    window = tk.Tk()

    window.title("Hospital Management Sysytem Installation Wizard")
    window.geometry("1000x600")

    F1 = tk.Frame(window, borderwidth = 6).pack(side = 'top', fill='x')
    F2 = tk.Frame(window, borderwidth = 6).pack(side = 'left', fill='y')
    F3 = tk.Frame(window, borderwidth = 6).pack(side = 'bottom', fill='x')

    L1 = tk.Label(F1, text = "Hospital Management System Installation Wizard", font = ("Arial", 40), bg = "pink", fg = "blue").pack(fill="x")
    L2 = tk.Label(F2, text = "Created by:- Apoorva Jadhav, Aayushi Kapoor & Anubhav Utkarsh [12th SCIENCE, session: 2020-21]", font = ("Arial", 20), fg = "purple").pack()
    L3 = tk.Label(F2, text = "=> Initially a databases will be created containing 4 tables as follows:", font = ("Arial", 20)).place(x=0,y=130)
    L4 = tk.Label(F2, text = "    1. Doctor table                                2. Patient table", font = ("Arial", 20)).place(x=0,y=155)

L0: 2 Col: 0
```

```
installation hospital.py - /Users/mac/Downloads/HMS Apoorva, Aayushi and Anubhav 12th Science/installation hospital.py (3.8.5)
l1 = tk.Label(window1, text = "Enter your MySQL password: ", font = ("Arial", 40), bg = "pink", fg = "blue").pack(fill='x')
E = tk.Entry(window1, width = 50, bg="black", fg="white", show="*")
E.pack(pady=10)
l2 = tk.Label(window1, text = "Click next to continue.", font = ("Arial", 20)).pack(side = 'left')
B1 = tk.Button(window1, text = "Next", font=("ComicSansMS", 20), fg="green", command = create1).place(x=940,y=500)

B2 = tk.Button(window1, text = "Back", font=("ComicSansMS", 20), fg="red", command = back1).place(x=10, y=500)

window1.resizable(False, False)
window1.mainloop()

def next1():
    window.destroy()
    create()

def w():
    global window
    window = tk.Tk()

    window.title("Hospital Management Sysstem Installation Wizard")
    window.geometry("1000x600")

    F1 = tk.Frame(window, borderwidth = 6).pack(side = 'top', fill='x')
    F2 = tk.Frame(window, borderwidth = 6).pack(side = 'left', fill='y')
    F3 = tk.Frame(window, borderwidth = 6).pack(side = 'bottom', fill='x')

    L1 = tk.Label(F1, text = "Hospital Management System Installation Wizard", font = ("Arial", 40), bg = "pink", fg = "blue").pack(fill="x")
    L2 = tk.Label(F2, text = "Created by:- Apoorva Jadhav, Aayushi Kapoor & Anubhav Utkarsh [12th SCIENCE, session: 2020-21]", font = ("Arial", 20), fg = "purple").pack()
    L3 = tk.Label(F2, text = "==> Initially a databases will be created containing 4 tables as follows:", font = ("Arial", 20)).place(x=0,y=130)
    L4 = tk.Label(F2, text = "    1. Doctor table          2. Patient table", font = ("Arial", 20)).place(x=0,y=155)
    L5 = tk.Label(F2, text = "    3. Diagnosise of patient table  4. Login table", font = ("Arial", 20)).place(x=0,y=180)
    L6 = tk.Label(F2, text = "==> MySQL must be installed before continuing.\nModules: mysql-connector, tkcalendar must be installed first ",font=("Arial", 20), fg = 'red').place(x=0,y=205)
    L7 = tk.Label(F2, text = "==> System requirements:", font = ("Arial", 20)).place(x=0,y=255)
    L8 = tk.Label(F2, text = "    (1) Storage minimum 1Gb, RAM minimum 1GB",font = ("Arial", 20)).place(x=0,y=280)
    L9 = tk.Label(F2, text = "    (2) Operating system :- MacOS Catalina",font = ("Arial", 20)).place(x=0,y=305)
    L10 = tk.Label(F3, text = "To continue, click next.", font = ("Arial", 20), fg = 'green').place(x=0,y=340)

    b1 = tk.Button(window, text="Next", font=("ComicSansMS", 20), fg="green", command = next1).place(x=940,y=500)
    b2 = tk.Button(window, text="Exit", font=("ComicSansMS", 20), fg="red", command = window.destroy).place(x=10, y=500)

    window.resizable(False, False)
    window.mainloop()

#main start
w()

Ln: 2 Col: 0
```

Hospital Management System Program(Codes):-



The screenshot shows a Python code editor window with the file 'HMS.py' open. The code is a Hospital Management System program written in Python using Tkinter for the graphical interface and mysql.connector for database interaction. The code includes functions for backtracking to the login screen, handling user input for creating a new account, and displaying error messages. The code is color-coded for readability, with syntax highlighting for keywords, comments, and strings.

```
1 __author__ = "APOORVA JADHAV, AAYUSHI KAPOOR & ANUBHAV UTKARSH"
2 __copyright__ = "Copyright (C) 2021 AAA"
3 __license__ = "Public Domain"
4 __version__ = "1.0"
5
6 import mysql.connector as mc
7 from mysql.connector import Error
8 import tkinter as tk
9 import tkinter.messagebox as tmb
10 from tkinter import ttk
11 from tkinter import *
12 from tkcalendar import Calendar, DateEntry
13
14 e19 = '1000-01-01 00:00:00'
15 e20 = '1000-01-01 00:00:00'
16 e21 = '1000-01-01 00:00:00'
17 #####
18
19 def backtolm_rg():
20     window5.destroy()
21     login_management()
22
23 def cdup():
24     du = e0.get()
25     dp = e00.get()
26     drp = e000.get()
27
28     q1 = "select username from login;"
29     query_ob = conn_ob.cursor()
30     query_ob.execute(q1)
31     r = query_ob.fetchall()
32     if du=='' or dp=='' or drp=='':
33         tmb.showinfo('error','Enter every field correctly !!!')
34     else:
35
36         q2 = "select * from login where username = '{0}';".format(du)
37         query_ob = conn_ob.cursor()
38         query_ob.execute(q2)
39         r2 = query_ob.fetchall()
40
41         if r2==[]:
42             if dp==drp:
43                 Q="insert into login values ('{0}', '{1}', 3, '{2}');".format(du, dp, did__)
44                 query_ob = conn_ob.cursor()
45                 query_ob.execute(Q)
46                 conn_ob.commit()
47                 tmb.showinfo('Completed', "Doctor id "+str(did__)+" created successfully.\nUsername: " + str(du) + "\nPassword: " + str(dp))
48                 window0.destroy()
49                 login_management()
```

12

Ln: 1 Col: 0

```
41     if r2==[]:
42         if dp==drp:
43             Q="insert into login values ('{0}', '{1}', 3, '{2}')".format(du, dp, did__)
44             query_ob = conn_ob.cursor()
45             query_ob.execute(Q)
46             conn_ob.commit()
47             tmb.showinfo('Completed', "Doctor id "+str(did__)+" created successfully.\nUsername: " + str(du) + "\nPassword: " + str(dp))
48             window0.destroy()
49             login_management()
50     else:
51         tmb.showinfo('error','Password doesn't match !!!')
52 else:
53     tmb.showinfo('error','Username not available !!!')
54
55
56
57
58 def exe_reg_doc():
59     global did__, e0, e00, e000, window0
60
61     query="select did from doctor;"
62     query_ob = conn_ob.cursor()
63     query_ob.execute(query)
64     rdid = query_ob.fetchall()
65
66     did__ = e3.get()
67     name_doc = e4.get()
68     name_doc = name_doc.capitalize()
69     speciality = e5.get()
70     speciality = speciality.capitalize()
71     d_gender = gen_doc.get()
72
73     lst=[] #result is nested tuple. This makes it easier.
74     for i in range(0,len(rdid)):
75         lst.append(rdid[i][0])
76
77     if (did__=="" or name_doc=="" or speciality=="" or d_gender=="" or d_gender=="Gender" or did__.isnumeric()==False):
78         tmb.showinfo("Error","Insert every field correctly !!!")
79
80     elif did__.isnumeric():
81         if int(did__) not in lst:
82             q1="insert into doctor values('{0}', '{1}', '{2}', '{3}')".format(did__,name_doc,speciality,d_gender)
83             query_ob = conn_ob.cursor()
84             query_ob.execute(q1)
85             conn_ob.commit()
86             window5.destroy()
87
88             window0 = tk.Tk()
89             window0.title("create doctor userpass")
```

```
87
88     window0 = tk.Tk()
89     window0.title("create doctor userpass")
90     window0.geometry("1000x600")
91
92     LL = tk.Label(window0, text = "MANAGEMENT", font= ("Arial",40), fg="dark blue").pack()
93     l5 = tk.Label(window0, text = "^^^^^ DO NOT CLOSE THE CURRENT WINDOW ^^^^", font=('Arial',20), fg='red').pack(pady=10)
94     l2 = tk.Label(window0, text = "Create username: ", font = ('Arial', 20), fg = 'blue').pack(pady=20)
95     e0 = tk.Entry(window0, width = 25, bg = 'black', fg = 'white')
96     e0.pack(pady=20)
97
98     l3 = tk.Label(window0, text = "Create password: ", font = ('Arial', 20), fg = 'blue').pack(pady=20)
99     e00 = tk.Entry(window0, width = 25, bg = 'black', fg = 'pink')
100    e00.pack(pady=20)
101
102    l4 = tk.Label(window0, text = "Re-enter password: ", font = ('Arial', 20), fg = 'blue').pack(pady=20)
103    e000 = tk.Entry(window0, width = 25, bg = 'black', fg = 'pink')
104    e000.pack(pady=20)
105
106    b1 = tk.Button(window0, text = "Next", font = ('Arial',20), fg = 'green', command = cdup).pack()
107
108    window0.resizable(False, False)
109    window0.mainloop()
110
111    #login_management()
112 else:
113     tmb.showinfo("Error", "Doctor ID already exists!!!")
114
115
116
117
118 def reg_doc():
119     global e3, e4, e5, gen_doc, window5
120     window5 = tk.Tk()
121     window5.title("create doctor id")
122     window5.geometry("1000x600")
123
124
125     LL = tk.Label(window5, text = "REGISTER DOCTOR", font= ("Arial",40), fg="dark blue").pack()
126
127     qqqq="select did from doctor order by did desc limit 1;"
128     query_ob.execute(qqqq)
129     r9=query_ob.fetchall()
130
131     if r9==[]:
132         L1 = tk.Label(window5, text = "**Last did value: 0", font = ("Arial",20), fg = "red").place(x=250, y=100)
133     else:
134         va = r9[0][0]
135         L2 = tk.Label(window5, text = "**Last did value:" + str(va), font = ("Arial",20), fg = "red").place(x=250, y=100)
```

```
130
131     if r9==[]:
132         L1 = tk.Label(window5, text = "**Last did value: 0", font = ("Arial",20), fg = "red").place(x=250, y=100)
133     else:
134         va = r9[0][0]
135         L2 = tk.Label(window5, text = "**Last did value:" + str(va), font = ("Arial",20), fg = "red").place(x=250, y=100)
136
137 L3 = tk.Label(window5, text = "Create doctor's id:", font = ("Arial",20)).place(x=250, y=150)
138 e3 = tk.Entry(window5, width = 20, bg = 'black', fg = 'white')
139 e3.place(x=550, y=150)
140
141 L4 = tk.Label(window5, text = "Enter the name of the doctor:", font = ("Arial",20), fg = 'blue').place(x=250, y=200)
142 e4 = tk.Entry(window5, width = 20, bg = 'black', fg = 'white')
143 e4.place(x=550, y=200)
144
145 L5 = tk.Label(window5, text = "Enter the speciality of the doctor:", font = ("Arial",20), fg = 'blue').place(x=250, y=250)
146 e5 = tk.Entry(window5, width = 20, bg = 'black', fg = 'white')
147 e5.place(x=550, y=250)
148
149 L6 = tk.Label(window5, text = "Enter the gender of doctor:", font = ("Arial",20), fg = 'blue').place(x=250, y=300)
150 gen_doc = ttk.Combobox(window5, width = 20)
151 gen_doc['values'] = ('Gender','M','F','O')
152 gen_doc.place(x=550, y=300)
153 gen_doc.current(0)
154
155 b1 = tk.Button(window5, text = "Next", font = ("Arial", 20), fg = "green", command = exe_reg_doc).pack(side = 'right')
156 b2 = tk.Button(window5, text = "Exit", font = ("Arial", 20), fg = "red", command = backtolm_rg).pack(side = 'left')
157
158
159 window5.resizable(False, False)
160 window5.mainloop()
161
162
163
164 def next21():
165     window2.destroy()
166     reg_doc()
167
168 ##########
169
170 def backtolm_dd():
171     window6.destroy()
172     login_management()
173
174
175 def del_name():
176     global conn_obj, e6
177     ds = e6.get()
178
```

HMS.py - /Users/mac/Downloads/HMS Apoorva, Aayushi and Anubhav 12th Science/HMS.py (3.8.5)

```
168 #####
169
170 def backtolm_dd():
171     window6.destroy()
172     login_management()
173
174
175 def del_name():
176     global conn_ob, e6
177     ds = e6.get()
178
179     query = "select id from login;"
180     query_ob = conn_ob.cursor()
181     query_ob.execute(query)
182     result = query_ob.fetchall()
183
184     lst=[]
185     for i in range(0,len(result)):
186         lst.append(result[i][0])
187
188     if ds == '':
189         tmbo.showinfo("Error","Enter doctor's id correctly!")
190     elif ds.isnumeric():
191         if int(ds) in lst:
192             q="delete from login where id='{0}'.format(ds)
193             query_ob = conn_ob.cursor()
194             query_ob.execute(q)
195             conn_ob.commit()
196             tmbo.showinfo("Deleted", "Registration of doctor id "+str(ds)+" deleted!!!")
197             window6.destroy()
198             login_management()
199         elif ds not in lst:
200             tmbo.showinfo("Error", "ID does not exist!!!")
201     else:
202         tmbo.showinfo("Error", "Incorrect format!")
203
204
205
206 def del_doc():
207     global e6, window6
208
209     window6 = tk.Tk()
210
211     window6.title("Delete doctor id")
212     window6.geometry("1000x600")
213
214     l1 = tk.Label(window6, text = "DELETE DOCTOR id", font= ("Arial",40), fg="dark blue").pack()
215     l2 = tk.Label(window6, text="Enter doctor ID: ", font=("Arial",25), fg="red").pack(pady=20)
216     e6 = tk.Entry(window6, width = 25, bg="black", fg="white")
```

```
205
206 def del_doc():
207     global e6, window6
208
209     window6 = tk.Tk()
210
211     window6.title("Delete doctor id")
212     window6.geometry("1000x600")
213
214     l1 = tk.Label(window6, text = "DELETE DOCTOR id", font= ("Arial",40), fg="dark blue").pack()
215     l2 = tk.Label(window6, text="Enter doctor ID: ", font=("Arial",25), fg="red").pack(pady=20)
216     e6 = tk.Entry(window6, width = 25, bg="black", fg="white")
217     e6.pack(pady=10)
218
219     b1 = tk.Button(window6, text="Next", font=("Arial",20), fg="green", command = del_name).pack(side="right")
220     b2 = tk.Button(window6, text="Back", font=("Arial",20), fg="red", command = backtolm_dd).pack(side="left")
221
222     window6.resizable(False, False)
223     window6.mainloop()
224
225 def next22():
226     window2.destroy()
227     del_doc()
228 ##########
229
230 def backtolm_vd():
231     window7.destroy()
232     login_management()
233
234 def done_vdr():
235     window8.destroy()
236     login_management()
237
238
239 def exe_view_doc_rep():
240     global e7, window8
241     d_id = e7.get()
242
243     if d_id.isnumeric():
244         q1 = "select dname from doctor where did={0}".format(d_id)
245         query_ob=conn_ob.cursor()
246         query_ob.execute(q1)
247         r1 = query_ob.fetchall()
248
249     if r1 == [] :
250         tmb.showinfo("Error", "Doctor with ID "+ str(d_id)+" is not registered")
251         window7.destroy()
252         login_management()
253
```

```
242
243     if d_id.isnumeric():
244         q1 = "select dname from doctor where did={0}".format(d_id)
245         query_ob=conn_ob.cursor()
246         query_ob.execute(q1)
247         r1 = query_ob.fetchall()
248
249     if r1 == []:
250         tmb.showinfo("Error", "Doctor with ID "+ str(d_id)+" is not registered")
251         window7.destroy()
252         login_management()
253
254     else:
255         #window7.destroy()
256         q2 = "select pid, disease, iop, status from d_patient natural join doctor where did='{0}';".format(d_id)
257         query_ob = conn_ob.cursor()
258         query_ob.execute(q2)
259         r2 = query_ob.fetchall()
260
261     if r2 == []:
262         tmb.showinfo("Null", "No Results found !!!")
263         window7.destroy()
264         login_management()
265
266     else:
267         window7.destroy()
268         window8 = tk.Tk()
269
270         window8.title("Results")
271         window8.geometry("1000x600")
272
273         #create a main frame
274         main_frame = tk.Frame(window8)
275         main_frame.pack(fill = 'both', expand = 1)
276
277         #create a canvas
278         my_canvas = tk.Canvas(main_frame)
279         my_canvas.pack(side = 'left', fill = 'both', expand = 1)
280
281         #add a scrollbar to canvas
282         sb = ttk.Scrollbar(main_frame, orient = 'vertical', command = my_canvas.yview)
283         sb.pack(side = 'right', fill = 'y')
284
285         # configure the canvas
286         my_canvas.configure(yscrollcommand = sb.set)
287         my_canvas.bind('<Configure>', lambda e: my_canvas.configure(scrollregion = my_canvas.bbox('all')))
288
289         #create another frame inside the canvas
290         second_frame = tk.Frame(my_canvas)
```

```
289     #create another frame inside the canvas
290     second_frame = tk.Frame(my_canvas)
291
292     #add that new frame to a window in the canva
293     my_canvas.create_window((0,0), window = second_frame, anchor="nw")
294
295     count = 0
296     for i in r2:
297         count += 1
298         a = i[0]
299         b = i[1]
300         c = i[2]
301         d = i[3]
302
303         l1 = tk.Label(second_frame, text = str(count) + ". pid: " + str(a) + ", \tdisease: " + str(b) + ", \tin/out-door: " + str(c) + ", \tstatus: " + str(d) +
304                     '\n~~~~~',font = ('Arial',20), fg = 'blue').pack()
305         b1 = tk.Button(window8, text = "Done", font=('Arial',20), fg='green', command = done_vdr).pack(side = 'bottom')
306
307
308         window8.resizable(False, False)
309         window8.mainloop()
310     else:
311         tmb.showinfo("Error", "Enter doctor ID correctly !!!")
312
313 def exe_view_doc_rep1():
314     window7.destroy()
315     exe_view_doc_rep()
316
317 def view_doc_rep():
318     global window7, e7
319
320     window7 = tk.Tk()
321
322     window7.title("View doctors report")
323     window7.geometry("1000x600")
324
325     l1 = tk.Label(window7, text = "VIEW DOCTOR'S REPORT", font= ("Arial",40), fg="dark blue").pack()
326     l2 = tk.Label(window7, text="Enter doctor ID to view report : ", font=("Arial",25), fg="red").pack(pady = 20)
327     e7 = tk.Entry(window7, width = 25, bg="black", fg="white")
328     e7.pack(pady = 20)
329
330     b1 = tk.Button(window7, text="Next", font=("Arial",20), fg="green", command = exe_view_doc_rep).pack(side="right")
331     b2 = tk.Button(window7, text="Back", font=("Arial",20), fg="red", command = backtolm_vd).pack(side="left")
332
333     window7.resizable(False, False)
334     window7.mainloop()
335
336 def next23():
337     window2.destroy()
```

```
336 def next23():
337     window2.destroy()
338     view_doc_rep()
339
340 ##########
341 def rtlm():
342     window10.destroy()
343     login_management()
344
345 def find_pat():
346     global e8, CI, conn_ob, window10
347     disease_ = e8.get()
348     st = CI.get()
349     disease = disease_.capitalize()
350
351 q1="select pid, pname, address from d_patient natural join patient where (disease='{0}') and (status='{1}');".format(disease,st)
352 q2="select count(pid) from d_patient where disease='{0}';".format(disease)
353
354 query_ob = conn_ob.cursor()
355
356 query_ob.execute(q1)
357 r1=query_ob.fetchall()
358
359 query_ob.execute(q2)
360 r2=query_ob.fetchall()
361
362 if disease_ =='' or st=='' or st=='Select choice':
363     tmb.showinfo("Error","Enter values correctly.")
364
365 elif r2[0][0]==0 or r1 == []:
366     tmb.showinfo("Report","No such patient exist!")
367
368 else:
369     window9.destroy()
370     window10 = tk.Tk()
371     window10.title("Patient's report")
372     window10.geometry("1000x600")
373
374     #create a main frame
375     main_frame = tk.Frame(window10)
376     main_frame.pack(fill = 'both', expand = 1)
377
378     #create a canvas
379     my_canvas = tk.Canvas(main_frame)
380     my_canvas.pack(side = 'left', fill = 'both', expand = 1)
381
382     #add a scrollbar to canvas
383     sb = ttk.Scrollbar(main_frame, orient = 'vertical', command = my_canvas.yview)
384     sb.pack(side = 'right', fill = 'v')
```

```
368 else:
369     window9.destroy()
370     window10 = tk.Tk()
371     window10.title("Patient's report")
372     window10.geometry("1000x600")
373
374     #create a main frame
375     main_frame = tk.Frame(window10)
376     main_frame.pack(fill = 'both', expand = 1)
377
378     #create a canvas
379     my_canvas = tk.Canvas(main_frame)
380     my_canvas.pack(side = 'left', fill = 'both', expand = 1)
381
382     #add a scrollbar to canvas
383     sb = ttk.Scrollbar(main_frame, orient = 'vertical', command = my_canvas.yview)
384     sb.pack(side = 'right', fill = 'y')
385
386     # configure the canvas
387     my_canvas.configure(yscrollcommand = sb.set)
388     my_canvas.bind('<Configure>', lambda e: my_canvas.configure(scrollregion = my_canvas.bbox('all')))
389
390     #create another frame inside the canvas
391     second_frame = tk.Frame(my_canvas)
392
393     #add that new frame to a window in the canva
394     my_canvas.create_window((0,0), window = second_frame, anchor="nw")
395
396
397     count = 0
398     for i in r1:
399         count += 1
400         c=i[0]
401         d=i[1]
402         e=i[2]
403         l1=tk.Label(second_frame, text = str(count) +". pid: "+str(c)+" , name: "+ str(d)+" , address: "+ str(e) +
404             '\n~~~~~', font = ('Arial',20), fg = 'blue').pack()
405
406
407     b1 = tk.Button(window10, text="Done", font=("Arial",20), fg="green", command = rtlm).pack(side="bottom")
408
409
410     window10.resizable(False, False)
411     window10.mainloop()
412
413
414
415 def backtolm_vpr():
416     window9.destroy()
```

```
415 def backtolm_vpr():
416     window9.destroy()
417     login_management()
418
419
420 def view_patient_rep():
421     global e8, CI, window9
422
423     window9 = tk.Tk()
424     window9.title("view patient report")
425     window9.geometry("1000x600")
426
427     l1 = tk.Label(window9, text = "VIEW PATIENT REPORT", font= ("Arial",40), fg="dark blue").pack()
428     l2 = tk.Label(window9, text="Enter patient's disease: ", font= ("Arial",25), fg="blue").pack()
429     e8 = tk.Entry(window9, width = 25, bg="black", fg="white")
430     e8.pack(pady = 20)
431
432     l3 = tk.Label(window9, text="Enter patient's status (C:cured, I:Infected): ", font= ("Arial",25), fg="red").pack()
433     CI = ttk.Combobox(window9, width=25)
434     CI['values']=('Select choice','C','I')
435     CI.pack()
436     CI.current(0)
437
438     b1 = tk.Button(window9, text = "Next", font = ("Arial",20), fg = "green", command = find_pat).pack(side="right")
439     b2 = tk.Button(window9, text = "Back", font = ("Arial",20), fg = "red", command = backtolm_vpr).pack(side="left")
440
441     window9.resizable(False, False)
442     window9.mainloop()
443
444
445 def next24():
446     window2.destroy()
447     view_patient_rep()
448
449 def logout_1():
450     window2.destroy()
451     login()
452
453
454
455 def login_management():
456     global window2
457     window2 = tk.Tk()
458     window2.title("login management")
459     window2.geometry("1000x600")
460
461     L1 = tk.Label(window2, text = "MANAGEMENT", font= ("Arial",40), fg="dark blue").pack()
462
463     L2 = tk.Label(window2, text = "1. Register a new doctor". font = ("Arial",20).fa = 'blue').place(x=300. v=100)
```

HMS.py - /Users/mac/Downloads/HMS Apoorva, Aayushi and Anubhav 12th Science/HMS.py (3.8.5)

```
455 def login_management():
456     global window2
457     window2 = tk.Tk()
458     window2.title("login management")
459     window2.geometry("1000x600")
460
461     L1 = tk.Label(window2, text = "MANAGEMENT", font= ("Arial",40), fg="dark blue").pack()
462
463     L2 = tk.Label(window2, text = "1. Register a new doctor", font = ("Arial",20),fg = 'blue').place(x=300, y=100)
464     b1 = tk.Button(window2, text ="1", font=("Arial", 20), fg="green", command = next21).place(x=700, y=100)
465
466     L3 = tk.Label(window2, text = "2. Remove a doctor", font =("Arial",20),fg = 'blue').place(x=300, y=150)
467     b2 = tk.Button(window2, text ="2", font=("Arial",20), fg="green", command = next22).place(x=700, y=150)
468
469     L4 = tk.Label(window2, text = "3. View monthly report of doctors ", font =("Arial",20),fg = 'blue').place(x=300, y=200)
470     b3 = tk.Button(window2, text ="3", font=("Arial",20), fg="green", command = next23).place(x=700, y=200)
471
472     L5 = tk.Label(window2, text = "4. View disease wise patient registration", font =("Arial",20),fg = 'blue').place(x=300, y=250)
473     b4 = tk.Button(window2, text ="4", font=("Arial",20), fg="green", command = next24).place(x=700, y=250)
474
475     b5 = tk.Button(window2, text ="LOGOUT", font=("Arial", 20), fg="red", command = logout_1).place(x=450, y=400)
476
477     window2.resizable(False, False)
478     window2.mainloop()
479
480 ######
481 def reg_pat():
482     pid___ = e9.get()
483     pat_name = e10.get()
484     pat_name.capitalize()
485     pat_gen = gen_pat.get()
486     bgr = blood_gr.get()
487     pat_city = e11.get()
488     pat_city.capitalize()
489
490     if (pid___=="" or pat_name=="" or pat_gen=="" or pat_gen=="Choose Gender" or bgr=="" or bgr=="Choose Blood Group" or pat_city=="" or pid___.isnumeric()==False):
491         tmb.showinfo("Error","Insert every field correctly !!!")
492
493     else:
494         query="select * from patient where pid='{0}';".format(pid___)
495         query_ob = conn_ob.cursor()
496         query_ob.execute(query)
497         rpid = query_ob.fetchall()
498
499     if rpid == []:
500         q1 = "insert into patient values ('{0}', '{1}', '{2}', '{3}', '{4}', now());".format(pid___,pat_name,pat_gen,pat_city,bgr)
501         query_ob = conn_ob.cursor()
502         query_ob.execute(q1)
503         conn_ob.commit()
```

```
493     else:
494         query="select * from patient where pid='{0}'.format(pid____)
495         query_ob = conn_ob.cursor()
496         query_ob.execute(query)
497         rpid = query_ob.fetchall()
498
499     if rpid == []:
500         q1 = "insert into patient values ('{0}', '{1}', '{2}', '{3}', '{4}', now()).format(pid____, pat_name, pat_gen, pat_city, bgr)
501         query_ob = conn_ob.cursor()
502         query_ob.execute(q1)
503         conn_ob.commit()
504
505         cq="create table d_pat_{+str(pid____)+}
506             "(entry_dates datetime , symptoms varchar(100), disease varchar(50) , diagnosis varchar(100), medicines varchar(100), iop char(1) , "+_
507             "admit_date datetime, discharge_date datetime , appointment_date datetime , check(iop in('i','o','I','O')));"
508         query_ob = conn_ob.cursor()
509         query_ob.execute(cq)
510         conn_ob.commit()
511
512
513         tmb.showinfo('created', 'Patient id '+str(pid____)+' registered successfully.')
514         window11.destroy()
515         login_receptionist()
516     else:
517         tmb.showinfo("Error", "Patient ID already exists!!!\nChange patient id to continue.")
518
519
520 def backtolr_rp():
521     window11.destroy()
522     login_receptionist()
523
524 def add_new_pat():
525     global window11, e9, e10, e11, gen_pat,blood_gr
526     window11 = tk.Tk()
527     window11.title("reg new pat")
528     window11.geometry("1000x600")
529
530     l1 = tk.Label(window11, text="REGISTER A NEW PATIENT", font = ("Arial", 40), fg = "dark blue").pack()
531
532     qp = "select pid from patient order by pid desc limit 1;"
533     query_ob = conn_ob.cursor()
534     query_ob.execute(qp)
535     rr = query_ob.fetchall()
536
537     if rr == []:
538         l2 = tk.Label(window11, text = "**Last pid value: 0", font = ("Arial",20), fg = "red").place(x=250, y=100)
539     else:
540         val = rr[0][0]
541         l3 = tk.Label(window11, text = "**Last pid value:" + str(val). font = ("Arial",20). fg = "red").place(x=250, v=100)
```

```
537 if rr == []:
538     l2 = tk.Label(window11, text = "**Last pid value: 0", font = ("Arial",20), fg = "red").place(x=250, y=100)
539 else:
540     val = rr[0][0]
541     l3 = tk.Label(window11, text = "**Last pid value:" + str(val), font = ("Arial",20), fg = "red").place(x=250, y=100)
542
543 l4 = tk.Label(window11, text ="Create patient ID : ",font = ("Arial", 20),fg = 'blue').place(x=250,y=150)
544 e9 = tk.Entry(window11, width = 25, bg="black", fg="white")
545 e9.place(x=500,y=150)
546
547 l5 = tk.Label(window11, text ="Enter patient name : ",font = ("Arial", 20),fg = 'blue').place(x=250,y=200)
548 e10 = tk.Entry(window11, width = 25, fg="white", bg="black")
549 e10.place(x=500,y=200)
550
551 l6 = tk.Label(window11, text ="Enter patient's gender : ",font = ("Arial", 20),fg = 'blue').place(x=250,y=250)
552 gen_pat = ttk.Combobox(window11, width = 25)
553 gen_pat['values'] = ('Choose Gender','M','F','O')
554 gen_pat.place(x=500, y=250)
555 gen_pat.current(0)
556
557 l7 = tk.Label(window11, text ="Enter Blood Group : ",font = ("Arial", 20), fg = 'blue').place(x=250,y=300)
558 blood_gr = ttk.Combobox(window11, width = 25)
559 blood_gr['values'] = ('Choose Blood Group','A+ve','A-ve','B+ve','B-ve','O+ve','O-ve','AB+ve','AB-ve')
560 blood_gr.place(x=500,y=300)
561 blood_gr.current(0)
562
563 l8 = tk.Label(window11, text ="Enter patient's city : ",font = ("Arial", 20),fg = 'blue').place(x=250,y=350)
564 e11 = tk.Entry(window11, width = 25, fg="white", bg="black")
565 e11.place(x=500,y=350)
566
567 b1 = tk.Button(window11, text = "Register", font=("Arial", 20), fg="green", command = reg_pat).place(x=915, y=400)
568 b2 = tk.Button(window11, text = "Back", font=("Arial", 20), fg="red", command = backtolr_rp).place(x=0, y=400)
569
570 window11.resizable(False, False)
571 window11.mainloop()
572
573
574 def next31():
575     window3.destroy()
576     add_new_pat()
577 ##########
578 def backtolr_ps():
579     window12.destroy()
580     login_receptionist()
581
582 def done_ps():
583     window13.destroy()
584     login_receptionist()
585
```

```
586 def run_pat_s():
587     name_pat = e12.get()
588     pg = gen_pat1.get()
589
590     if (name_pat == '' or pg == '' or pg == 'Choose Gender'):
591         tmbo.showinfo("Error", "Insert every field correctly !!!")
592     else:
593         q5="select pid from patient where pname like '%{0}%' and gender = '{1}'".format(name_pat,pg)
594         query_ob = conn_ob.cursor()
595         query_ob.execute(q5)
596         res=query_ob.fetchall()
597
598     if res==[]:
599         tmbo.showinfo("s result","No patient found !!!")
600     else:
601         window12.destroy()
602         global window13
603         window13 = tk.Tk()
604         window13.title("Result")
605         window13.geometry("1000x600")
606
607         #create a main frame
608         main_frame = tk.Frame(window13)
609         main_frame.pack(fill = 'both', expand = 1)
610
611         #create a canvas
612         my_canvas = tk.Canvas(main_frame)
613         my_canvas.pack(side = 'left', fill = 'both', expand = 1)
614
615         #add a scrollbar to canvas
616         sb = ttk.Scrollbar(main_frame, orient = 'vertical', command = my_canvas.yview)
617         sb.pack(side = 'right', fill = 'y')
618
619         # configure the canvas
620         my_canvas.configure(yscrollcommand = sb.set)
621         my_canvas.bind('<Configure>', lambda e: my_canvas.configure(scrollregion = my_canvas.bbox('all')))
622
623         #create another frame inside the canvas
624         second_frame = tk.Frame(my_canvas)
625
626         #add that new frame to a window in the canva
627         my_canvas.create_window((0,0), window = second_frame, anchor="nw")
628
629         count = 0
630         for j in range(len(res)):
631             count += 1
632             p = res[j][0]
633
634             l2 = tk.Label(second_frame, text = str(count) + " Patient's id (DTN) : " + str(p) + "
```

```
624 second_frame = tk.Frame(my_canvas)
HMS.py - /Users/mac/Downloads/HMS Apoorva, Aayushi and Anubhav 12th Science/HMS.py (3.8.5)
625
626     #add that new frame to a window in the canvas
627     my_canvas.create_window((0,0), window = second_frame, anchor="nw")
628
629     count = 0
630     for j in range(len(res)):
631         count += 1
632         p = res[j][0]
633
634         l2 = tk.Label(second_frame, text = str(count) + ". Patient's id (PID):   " + str(p) +
635                     '\n~~~~~', font = ('Arial',20), fg = 'blue').pack()
636         b = tk.Button(window13, text = "Done", font=("Arial", 20), fg="green", command = done_ps).place(x=450, y=550)
637
638         window13.resizable(False, False)
639         window13.mainloop()
640
641
642 def pat_s():
643     global window12, e12, gen_pat1
644     window12 = tk.Tk()
645     window12.title("search a patient")
646     window12.geometry("1000x600")
647
648     l1 = tk.Label(window12, text="SEARCH A PATIENT",font = ("Arial", 40), fg = "dark blue").pack()
649
650     l2 = tk.Label(window12, text = "Enter patient's name : ", font = ("Arial",20), fg = "blue").pack(pady=30)
651     e12 = tk.Entry(window12, width = 25, fg="white", bg="black")
652     e12.pack(pady=10)
653
654     l3 = tk.Label(window12, text = "Enter patient's blood group: ", font = ("Arial",20), fg = "blue").pack(pady=20)
655     gen_pat1 = ttk.Combobox(window12, width = 25)
656     gen_pat1['values'] = ('Choose Gender','M','F','O')
657     gen_pat1.pack(pady=10)
658     gen_pat1.current(0)
659
660     b1 = tk.Button(window12, text = "Search", font=("Arial", 20), fg="green", command = run_pat_s).place(x=915, y=400)
661     b2 = tk.Button(window12, text = "Back", font=("Arial", 20), fg="red", command = backtolr_ps).place(x=0, y=400)
662
663     window12.resizable(False, False)
664     window12.mainloop()
665
666
667 def next32():
668     window3.destroy()
669     pat_s()
670 ##########
671 def cfa():
672     c_a = e13.get()
673
```

HMSS.py - /Users/mac/Downloads/HMS Apoorva, Aayushi and Anubhav 12th Science/HMS.py (3.8.5)

```
555 windows.destroy()
556
557 pat_s()
558 #####
559
560 def cfa():
561     c_a = e13.get()
562
563     if c_a == '':
564         tmb.showinfo('error','Enter patient id !!!')
565     elif c_a.isnumeric()==False:
566         tmb.showinfo('error','Enter patient correctly,\ncharecters like alphabets and special characters are not allowed !!!')
567     else:
568         q10 = "select * from patient where pid = " + str(c_a) + str(';')
569         query_ob.execute(q10)
570         r3 = query_ob.fetchall()
571
572         if r3==[]:
573             tmb.showinfo('error','Patient not registered !!!')
574             window14.destroy()
575             login_receptionist()
576
577         else:
578             q11 = "select date(appointment_date) from d_pat_"+str(c_a)+" order by appointment_date desc limit 1;"
579             query_ob.execute(q11)
580             r4 = query_ob.fetchall()
581
582             if r4==[]:
583                 tmb.showinfo('result','No appoinments !!!')
584                 window14.destroy()
585                 login_receptionist()
586             elif r4[0][0]=="1000-01-01 00:00:00":
587                 tmb.showinfo('result','No appoinments !!!')
588                 window14.destroy()
589                 login_receptionist()
590             else:
591                 tmb.showinfo('result','Pid:  ' + str(c_a) + "\nLast Appoiment date:  " + str(r4[0][0]))
592                 window14.destroy()
593                 login_receptionist()
594
595
596
597 def backtolr_ca():
598     window14.destroy()
599     login_receptionist()
600
601
602 def ca():
603     global window14, e13
604     window14 = tk.Tk()
605     window14.title("check appoinment")
606     window14.geometry("1000x600")
607
608     l1 = tk.Label(window14, text = "SEARCH FOR APPOINTMENT", font = ("Arial", 10), fg = "dark blue").pack()
```

```
711 def ca():
712     global window14, e13
713     window14 = tk.Tk()
714     window14.title("check appoinment")
715     window14.geometry("1000x600")
716
717     l1 = tk.Label(window14, text = "SEARCH FOR APPOINTMENT",font = ("Arial", 40), fg = "dark blue").pack()
718
719     l2 = tk.Label(window14, text = "Enter patient's id (PID no):", font = ('Arial',20), fg = 'blue').pack(pady=40)
720     e13 = tk.Entry(window14, width = 25, bg='black', fg='white')
721     e13.pack(pady=10)
722
723     l3 = tk.Label(window14, text = "", font = ('Arial',20), fg = 'blue').pack(pady=40)
724
725     b1 = tk.Button(window14, text='Check', font=('Arial',20), fg='green', command = cfa).place(x=915, y=400)
726     b2 = tk.Button(window14, text = "Back", font=("Arial", 20), fg="red", command = backtolr_ca).place(x=0, y=400)
727
728     window14.resizable(False, False)
729     window14.mainloop()
730
731 def next33():
732     window3.destroy()
733     ca()
734
735 def logout_2():
736     window3.destroy()
737     login()
738
739 def login_receptionist():
740     global window3
741     window3 = tk.Tk()
742     window3.title("Welcome to Hospital Management Sysytem")
743     window3.geometry("1000x600")
744
745     LL = tk.Label(window3, text = "RECEPTIONIST", font = ('Arial', 40), fg = 'dark blue').pack()
746
747     l1 = tk.Label(window3, text="1. Register a new patient ",font = ("Arial", 20),fg = 'blue').place(x=200,y=100)
748     b1 = tk.Button(window3, text ="1", font=("Arial", 20), fg="green", command = next31).place(x=700, y=100)
749
750     l2 = tk.Label(window3, text="2. Search for a new patient ",font = ("Arial", 20),fg = 'blue').place(x=200,y=150)
751     b2 = tk.Button(window3, text ="2", font=("Arial", 20), fg="green", command = next32).place(x=700, y=150)
752
753     l3 = tk.Label(window3, text="3. Check for Appointment ",font = ("Arial", 20),fg = 'blue').place(x=200,y=200)
754     b3 = tk.Button(window3, text ="3", font=("Arial", 20), fg="green", command = next33).place(x=700, y=200)
755
756     b4 = tk.Button(window3, text ="LOGOUT", font=("Arial", 20), fg="red", command = logout_2).place(x=450, y=350)
757
758     window3.resizable(False, False)
759     window3.mainloop()
```

```
761 #####  
762 def cont_dp():  
763     global da, da1, da2  
764     pid_ = id_pid  
765     yid = id_did  
766     symp = e15.get()  
767     symp.capitalize()  
768     bimari = e16.get()  
769     bimari.capitalize()  
770     dia = e17.get()  
771     dia.capitalize()  
772     med = e18.get()  
773     med.capitalize()  
774     iop = I0.get()  
775     iop.capitalize()  
776     ad = e19  
777     dd = e20  
778     apd = e21  
779     status = IC.get()  
780     status.capitalize()  
781     if (symp == '' or bimari == '' or dia == '' or med == '' or iop == 'Select choice' or status == 'Select choice'  
782         or pid_ == '' or yid == '' or (ad=='1000-01-01 00:00:00' and dd=='1000-01-01 00:00:00' and apd=='1000-01-01 00:00:00')):#or (ad==dd)  
783         tmb.showinfo('error', "Enter fields correctly !!!")  
784     elif dd == '1000-01-01 00:00:00':  
785         qq = ("insert into d_pat_"+str(pid_)+"(entry_dates,symptoms,disease,diagnosis,medicines,iop,admit_date,appointment_date) values("+  
786             "CURRENT_TIMESTAMP(),'{0}', '{1}', '{2}', '{3}', '{4}', '{5}', '{6}');".format(symp,bimari,dia,med,iop,ad,apd))  
787         query_ob.execute(qq)  
788         conn_ob.commit()  
789  
790         qqq="insert into d_patient values ('{0}', '{1}', '{2}', '{3}', '{4}')".format(pid_,yid,bimari,iop,status)  
791         query_ob.execute(qqq)  
792         conn_ob.commit()  
793  
794         tmb.showinfo('Done','Diagnose recorded !!!')  
795  
796         window16.destroy()  
797         login_doctor(yid)  
798  
799     elif ad == '1000-01-01 00:00:00':  
800         qq = ("insert into d_pat_"+str(pid_)+"(entry_dates,symptoms,disease,diagnosis,medicines,iop,discharge_date,appoinment_date) values"+  
801             "(CURRENT_TIMESTAMP(),'{0}', '{1}', '{2}', '{3}', '{4}', '{5}', '{6}');".format(symp,bimari,dia,med,iop,dd,apd))  
802         query_ob.execute(qq)  
803         conn_ob.commit()  
804  
805         qqq="insert into d_patient values ('{0}', '{1}', '{2}', '{3}', '{4}')".format(pid_,yid,bimari,iop,status)  
806         query_ob.execute(qqq)  
807         conn_ob.commit()  
808  
809         tmb.showinfo('Done','Diagnose recorded !!!')
```

```
806     query_ob.execute(qqq)
807     conn_ob.commit()
808
809     tmb.showinfo('Done','Diagnose recorded !!!')
810
811     window16.destroy()
812     login_doctor(yid)
813
814 elif ad == '1000-01-01 00:00:00' and apd == '1000-01-01 00:00:00':
815     qq = ("insert into d_pat_"+str(pid_)+(entry_dates,symptoms,disease,diagnosis,medicines,iop,discharge_date) values"+
816           "(CURRENT_TIMESTAMP(),'{0}', '{1}', '{2}', '{3}', '{4}', '{5}');".format(symp,bimari,dia,med,iop,dd))
817     query_ob.execute(qq)
818     conn_ob.commit()
819
820     tmb.showinfo('Done','Diagnose recorded !!!')
821
822     qqq="insert into d_patient values ('{0}', '{1}', '{2}', '{3}', '{4}')".format(pid_,yid,bimari,iop,status)
823     query_ob.execute(qqq)
824     conn_ob.commit()
825
826     window16.destroy()
827     login_doctor(yid)
828
829 else:
830     tmb.showinfo('error','DATES ERROR')
831
832
833 def calendar_view():
834     global da, e19
835     def print_sel():
836         global da, e19
837         da = cal.selection_get()
838         e19 = da
839         l11.config(text = da)
840         top.destroy()
841
842     def nod():
843         global da, e19
844         da = '1000-01-01 00:00:00'
845         e19 = da
846         l11.config(text = 'None')
847         top.destroy()
848
849     top = tk.Toplevel(window16)
850
851     cal = Calendar(top,
852                   font="Arial 14", selectmode='day',
853                   cursor="hand1")
854     cal.pack(fill="both", expand=True)
```

```
850
851     cal = Calendar(top,
852                     font="Arial 14", selectmode='day',
853                     cursor="hand1")
854     cal.pack(fill="both", expand=True)
855     b1 = ttk.Button(top, text="Ok", command=print_sel).pack(side = 'right')
856     b2 = ttk.Button(top, text="None", command=nod).pack(side = 'left')
857
858 def calendar_view1():
859     global da1
860     def print_sel():
861         global da1, e20
862         da1 = cal.selection_get()
863         e20 = da1
864         l12.config(text = da1)
865         top.destroy()
866
867     def nod():
868         global da1, e20
869         da1 = '1000-01-01 00:00:00'
870         e20 = da1
871         l12.config(text = 'None')
872         top.destroy()
873
874     top = tk.Toplevel(window16)
875
876     cal = Calendar(top,
877                     font="Arial 14", selectmode='day',
878                     cursor="hand1")
879     cal.pack(fill="both", expand=True)
880     b1 = ttk.Button(top, text="Ok", command=print_sel).pack(side = 'right')
881     b2 = ttk.Button(top, text="None", command=nod).pack(side = 'left')
882
883
884 def calendar_view2():
885     global da2
886     def print_sel():
887         global da2, e21
888         da2 = cal.selection_get()
889         e21 = da2
890         l13.config(text = da2)
891         top.destroy()
892
893     def nod():
894         global da2, e21
895         da2 = '1000-01-01 00:00:00'
896         e21 = da2
897         l13.config(text = 'None')
898         top.destroy()
```

```
899
900 top = tk.Toplevel(window16)
901
902 cal = Calendar(top,
903                 font="Arial 14", selectmode='day',
904                 cursor="hand1")
905 cal.pack(fill="both", expand=True)
906 b1 = ttk.Button(top, text="Ok", command=print_sel).pack(side = 'right')
907 b2 = ttk.Button(top, text="None", command=nod).pack(side = 'left')
908
909
910 def exe_dp():
911     global yid, id_did
912     yid = id_did
913     window15.destroy()
914     global window16, e15, e16, e17, e18, e19, e20, e21, I0, IC, l11, l12, l13
915     window16 = tk.Tk()
916     window16.title("diagnosis patient")
917     window16.geometry("1000x600")
918
919     s = ttk.Style(window16)
920     s.theme_use('clam')
921
922     L1 = tk.Label(window16, text = "DOCTOR", font = ("Arial",40), fg="dark blue").pack()
923
924     l2 = tk.Label(window16, text = "1. Enter symptoms of patient: ", font = ("Arial", 20),fg = 'blue').place(x = 170, y=80)
925     e15 = tk.Entry(window16, width = 25, bg = 'black', fg = 'white')
926     e15.place(x = 630, y=80)
927
928
929     l3 = tk.Label(window16, text = "2. Enter disease of patient: ", font = ("Arial", 20),fg = 'blue').place(x = 170, y=120)
930     e16 = tk.Entry(window16, width = 25, bg = 'black', fg = 'white')
931     e16.place(x = 630, y=120)
932
933     l4 = tk.Label(window16, text = "3. Enter diagnosis done or suggested: ", font = ("Arial", 20),fg = 'blue').place(x = 170, y=160)
934     e17 = tk.Entry(window16, width = 25, bg = 'black', fg = 'white')
935     e17.place(x = 630, y=160)
936
937     l5 = tk.Label(window16, text = "4. Enter medicines prescribed: ", font = ("Arial", 20),fg = 'blue').place(x = 170, y=200)
938     e18 = tk.Entry(window16, width = 25, bg = 'black', fg = 'white')
939     e18.place(x = 630, y=200)
940
941     l6 = tk.Label(window16, text = "5. Enter whether patient is indoor or outdoor [I,O]: ", font = ("Arial", 20),fg = 'blue').place(x = 170, y=240)
942     I0 = ttk.Combobox(window16, width=25)
943     I0['values']=('Select choice','I','O')
944     I0.place(x = 630, y=240)
945     I0.current(0)
946
947     l7 = tk.Label(window16, text = "6. Enter ADMIT date (YYYY-MM-DD): ", font = ("Arial", 20),fg = 'blue').place(x = 170, y=280)
```

```
945 IO.current(0)
946
947 l7 = tk.Label(window16, text = "6. Enter ADMIT date (YYYY-MM-DD): ", font = ("Arial", 20),fg = 'blue').place(x = 170, y=280)
948 b1 = ttk.Button(window16, text='Select date', command=calendar_view).place(x = 630, y=280)
949 l11 = tk.Label(window16, text='', font = ("Arial", 20),fg = 'blue')
950 l11.place(x = 800, y=280)
951
952 l8 = tk.Label(window16, text = "7. Enter DISCHARGE date (YYYY-MM-DD): ", font = ("Arial", 20),fg = 'blue').place(x = 170, y=320)
953 b2 = ttk.Button(window16, text='Select date', command=calendar_view1).place(x = 630, y=320)
954 l12 = tk.Label(window16, text='', font = ("Arial", 20),fg = 'blue')
955 l12.place(x = 800, y=320)
956
957 l9 = tk.Label(window16, text = "8. Enter next APPOINTMENT date (YYYY-MM-DD): ", font = ("Arial", 20),fg = 'blue').place(x = 170, y=360)
958 b3 = ttk.Button(window16, text='Select date', command=calendar_view2).place(x = 630, y=360)
959 l13 = tk.Label(window16, text='', font = ("Arial", 20),fg = 'blue')
960 l13.place(x = 800, y=360)
961
962 l10 = tk.Label(window16, text = "9. Cured-'C'/Infected-'I': ", font = ("Arial", 20),fg = 'blue').place(x = 170, y=400)
963 IC = ttk.Combobox(window16, width=25)
964 IC['values']=('Select choice','C','I')
965 IC.place(x = 630, y=400)
966 IC.current(0)
967
968 b4 = tk.Button(window16, text = "Continue", font = ('Arial', 20), fg = 'green', command = cont_dp).pack(side = 'bottom')
969
970
971 window16.resizable(False, False)
972 window16.mainloop()
973
974 def backtold():
975     window15.destroy()
976     login_doctor(id_did)
977
978 def OK():
979     window15.destroy()
980     login_doctor(id_did)
981
982 def dp():
983     global pid_
984     pid_ = e14.get()
985     window4.destroy()
986
987     global window15, e15
988     window15 = tk.Tk()
989     window15.title("login doctor")
990     window15.geometry("1000x600")
991
992 L1 = tk.Label(window15, text = "DOCTOR", font = ("Arial",40), fg="dark blue").pack()
```

HMS.py - /Users/mac/Downloads/HMS Apoorva, Aayushi and Anubhav 12th Science/HMS.py (3.8.5)

```
983 def dp():
984     global pid_
985     pid_ = e14.get()
986     window4.destroy()
987
988     global window15, e15
989     window15 = tk.Tk()
990     window15.title("login doctor")
991     window15.geometry("1000x600")
992
993     L1 = tk.Label(window15, text = "DOCTOR", font = ("Arial",40), fg="dark blue").pack()
994
995     q6 = "select * from patient where pid={0};".format(pid_)
996     query_ob.execute(q6)
997     r = query_ob.fetchall()
998
999     l1 = tk.Label(window15, text = "Patient id (pid): " + str(r[0][0]) +
1000         "\nPatient's name: " + str(r[0][1]) +
1001         "\nPatient's gender: " + str(r[0][2]) +
1002         "\nPatient's blood group: " + str(r[0][4]), font = ('Arial',20), fg = 'orange').pack()
1003
1004     q8 = "select * from d_pat_" + str(pid_) + ";"
1005     query_ob.execute(q8)
1006     r3 = query_ob.fetchall()
1007
1008     l2 = tk.Label(window15, text = "Patient history: " + str(len(r3)) + "*", font = ('Arial',20), fg = 'brown').pack()
1009
1010     #create a main frame
1011     main_frame = tk.Frame(window15)
1012     main_frame.pack(fill = 'both', expand = 1)
1013
1014     #create a canvas
1015     my_canvas = tk.Canvas(main_frame)
1016     my_canvas.pack(side = 'left', fill = 'both', expand = 1)
1017
1018     #add a scrollbar to canvas
1019     sb = ttk.Scrollbar(main_frame, orient = 'vertical', command = my_canvas.yview)
1020     sb.pack(side = 'right', fill = 'y')
1021
1022     # configure the canvas
1023     my_canvas.configure(yscrollcommand = sb.set)
1024     my_canvas.bind('<Configure>', lambda e: my_canvas.configure(scrollregion = my_canvas.bbox('all')))
1025
1026     #create another frame inside the canvas
1027     second_frame = tk.Frame(my_canvas)
1028
1029     #add that new frame to a window in the canva
1030     my_canvas.create_window((0,0), window = second_frame, anchor="nw")
1031
```

HMS.py - /Users/mac/Downloads/HMS Apoorva, Aayushi and Anubhav 12th Science/HMS.py (3.8.5)

```
1023 my_canvas.configure(yscrollcommand = sb.set)
1024 my_canvas.bind('<Configure>', lambda e: my_canvas.configure(scrollregion = my_canvas.bbox('all')))
1025
1026 #create another frame inside the canvas
1027 second_frame = tk.Frame(my_canvas)
1028
1029 #add that new frame to a window in the canva
1030 my_canvas.create_window((0,0), window = second_frame, anchor="nw")
1031
1032
1033
1034 for i in range(len(r3)):
1035     l3 = tk.Label(second_frame, text = "\n^n|History: " + str(i+1) +
1036                 "|^ ,\n|Entry date: " + str(r3[i][0]) +
1037                 "| ,\n|Symptoms: " + str(r3[i][1]) +
1038                 "| ,\n|Disease: " + str(r3[i][2]) +
1039                 "| ,\n|Diagnosis: " + str(r3[i][3]) +
1040                 "| ,\n|Medicines given: " + str(r3[i][4]) +
1041                 "| ,\n|Patient admitted or not: " + str(r3[i][5]) +
1042                 "| ,\n|Admit date (if admited): " + str(r3[i][6]) +
1043                 "| ,\n|Discharge date: " + str(r3[i][7]) +
1044                 "| ,\n|Appoinment date: " + str(r3[i][8]) +
1045                 '| ,\n~~~~~',
1046                 font = ('Arial',20), fg = 'blue').pack()
1047
1048 b1 = tk.Button(window15, text = 'Continue', font = ('Arial',20), fg = 'green', command = exe_dp).pack(side = 'bottom')
1049 b2 = tk.Button(window15, text = 'Back', font = ('Arial',20), fg = 'red', command = OK).pack(side = 'bottom')
1050
1051 window15.resizable(False, False)
1052 window15.mainloop()
1053
1054
1055 def next41():
1056     global id_pid
1057     id_pid = e14.get()
1058     if id_pid == "":
1059         tmb.showinfo('error','Enter patient id !!!')
1060     elif id_pid.isnumeric()==False:
1061         tmb.showinfo('error','Enter patient id correctly,\ncharacter like alphabets and special character are not allowed !!!')
1062     else:
1063         q6 = "select * from patient where pid={0};".format(id_pid)
1064         query_ob.execute(q6)
1065         r = query_ob.fetchall()
1066         if r == []:
1067             tmb.showinfo('error','Patient is not registered !!!')
1068         else:
1069             dp()
1070
1071
```

```
1071
1072 def logout_doctor():
1073     window4.destroy()
1074     login()
1075
1076 def login_doctor(x):
1077     global window4, id_did, e14
1078     window4 = tk.Tk()
1079     window4.title("Welcome to Hospital Management Sysytem")
1080     window4.geometry("1000x600")
1081
1082     id_did = x
1083
1084     L1 = tk.Label(window4, text = "DOCTOR", font= ("Arial",40), fg="dark blue").pack()
1085
1086     l2 = tk.Label(window4, text = 'Diagnose the patient:', font = ('Arial',20), fg='blue').pack(pady=30)
1087     l2 = tk.Label(window4, text = "Enter patient's id: ", font = ('Arial',20), fg='blue').pack(pady=10)
1088     e14 = tk.Entry(window4, width = 25, bg = 'black', fg = 'white')
1089     e14.pack()
1090
1091
1092     b1 = tk.Button(window4, text ="Next", font =("Arial",20), fg = 'green', command = next41).pack(side = 'right')
1093     b2 = tk.Button(window4, text ="LOGOUT", font=( "Arial", 20), fg="red", command = logout_doctor).pack(side = 'left')
1094
1095     window4.resizable(False, False)
1096     window4.mainloop()
1097
1098 def checkup(): #username password check
1099     global e1, e2, conn_ob, query_ob, u
1100     u = e1.get()
1101     ps = e2.get()
1102     q99 = "select * from login"
1103     query_ob = conn_ob.cursor()
1104     query_ob.execute(q99)
1105     r99 = query_ob.fetchall()
1106
1107     if (u != '' and ps != ''):
1108         for i in r99:
1109             if i[0] == u:
1110                 if i[1] == ps:
1111                     role = i[2]
1112                     if role == 1:
1113                         tmb.showinfo("login_management","Welcome "+u)
1114                         window1.destroy()
1115                         login_management()
1116                         break
1117                     elif role == 2:
1118                         tmb.showinfo("login_receptionist","Welcome "+u)
1119                         window1.destroy()
```

HMS.py - /Users/mac/Downloads/HMS Apoorva, Aayushi and Anubhav 12th Science/HMS.py (3.8.5)

```
109/
1098 def checkup(): #username password check
1099     global e1, e2, conn_ob, query_ob, u
1100     u = e1.get()
1101     ps = e2.get()
1102     q99 = "select * from login"
1103     query_ob = conn_ob.cursor()
1104     query_ob.execute(q99)
1105     r99 = query_ob.fetchall()
1106
1107     if (u != '' and ps != ''):
1108         for i in r99:
1109             if i[0] == u:
1110                 if i[1] == ps:
1111                     role = i[2]
1112                     if role == 1:
1113                         tmb.showinfo("login_management","Welcome "+u)
1114                         window1.destroy()
1115                         login_management()
1116                         break
1117                     elif role == 2:
1118                         tmb.showinfo("login_receptionist","Welcome "+u)
1119                         window1.destroy()
1120                         login_receptionist()
1121                         break
1122                     elif role==3:
1123                         id_ = i[3]
1124                         tmb.showinfo("login_doctor","Welcome "+u)
1125                         window1.destroy()
1126                         login_doctor(id_)
1127                         break
1128                 else:
1129                     tmb.showinfo("Error","INCORRECT USERNAME OR PASSWORD !!!")
1130
1131             else:
1132                 tmb.showinfo("Error","Enter both USERNAME and PASSWORD !!!")
1133
1134
1135 def back1():
1136     window1.destroy()
1137     w()
1138
1139
1140 def login(): #username password enter
1141     global e1, e2, window1
1142     window1 = tk.Tk()
1143
1144     window1.title("Login window")
1145     window1.geometry("1000x600")
1146
```

A screenshot of a Mac OS X terminal window titled "HMS.py - /Users/mac/Downloads/HMS Apoorva, Aayushi and Anubhav 12th Science/HMS.py (3.8.5)". The window contains a large amount of Python code, line-numbered from 1128 to 1176. The code is color-coded: purple for strings, blue for variables, green for comments, and orange for keywords like 'else' and 'except'. The code defines functions for handling user input, creating a login window with Tkinter, and connecting to a MySQL database using MySQL Connector. It also includes error handling and message boxes.

```
1128     else:
1129         tmbo.showinfo("Error","INCORRECT USERNAME OR PASSWORD !!!!")
1130
1131     else:
1132         tmbo.showinfo("Error","Enter both USERNAME and PASSWORD !!!")
1133
1134
1135 def back1():
1136     window1.destroy()
1137     w()
1138
1139
1140 def login(): #username password enter
1141     global e1, e2, window1
1142     window1 = tk.Tk()
1143
1144     window1.title("Login window")
1145     window1.geometry("1000x600")
1146
1147     l1 = tk.Label(window1, text = "WELCOME TO HOSPITAL MANAGEMENT SYSTEM", font = ("Arial", 20), fg = "purple").pack(pady = 20)
1148     l2 = tk.Label(window1, text = "Username: ", font = ("Arial", 20), fg = "blue").pack(pady = 10)
1149     e1 = tk.Entry(window1, width = 25, bg = "black", fg = "red")
1150     e1.pack()
1151     l3 = tk.Label(window1, text = "Password: ", font = ("Arial", 20), fg = "blue").pack(pady = 10)
1152     e2 = tk.Entry(window1, width = 25, bg = "black", fg = "white", show = "*")
1153     e2.pack()
1154     b1 = tk.Button(window1, text = "Login", font = ("Arial", 20), fg = "green", command = checkup).pack(pady=20)
1155     b2 = tk.Button(window1, text = "Back", font = ("Arial", 20), fg = "red", command = back1).pack(side = 'left')
1156
1157
1158     window1.resizable(False, False)
1159     window1.mainloop()
1160
1161
1162
1163 def con(): #connecting python to mysql
1164     global E, conn_ob
1165     p = E.get()
1166     try:
1167         conn_ob=mc.connect(host="localhost",database="hospital",user="root",password=p)
1168         query_ob=conn_ob.cursor()
1169         if (conn_ob.is_connected()):
1170             tmbo.showinfo("Connected","CONNECTED, click OK to continue.")
1171             window.destroy()
1172             login()
1173
1174
1175     except Error as e:
1176         tmbo.showinfo("Error occured",e)
```

HMS.py - /Users/mac/Downloads/HMS Apoorva, Aayushi and Anubhav 12th Science/HMS.py (3.8.5)

```
1156
1157
1158     window1.resizable(False, False)
1159     window1.mainloop()
1160
1161
1162
1163 def con(): #connecting python to mysql
1164     global E, conn_ob
1165     p = E.get()
1166     try:
1167         conn_ob=mc.connect(host="localhost",database="hospital",user="root",password=p)
1168         query_ob=conn_ob.cursor()
1169         if (conn_ob.is_connected()):
1170             tmb.showinfo("Connected","CONNECTED, click OK to continue.")
1171             window.destroy()
1172             login()
1173
1174
1175     except Error as e:
1176         tmb.showinfo("Error occured",e)
1177
1178
1179
1180 #main start
1181
1182 def w():
1183
1184     global window, E
1185     window = tk.Tk()
1186
1187     window.title("Welcome to Hospital Management Sysytem")
1188     window.geometry("1000x600")
1189
1190     LL = tk.Label(window, text = "HOSPITAL MANAGEMENT SYSTEM", font = ("Arial", 40), bg = "pink", fg = "blue").pack(fill="x")
1191
1192     L1 = tk.Label(window, text = "Enter MySql password to initialize.", font = ("Arial", 20), fg = "blue").pack(pady = 170)
1193     E = tk.Entry(window, width = 25, bg = 'black', fg = 'white', show = '*')
1194     E.pack(pady = 20)
1195     B1 = tk.Button(window, text = "Next", font = ("Arial", 20), fg = "green", command = con).pack(side = 'right')
1196     B2 = tk.Button(window, text = "Exit", font = ("Arial", 20), fg = "red", command = window.destroy).pack(side = 'left')
1197
1198
1199     window.resizable(False, False)
1200     window.mainloop()
1201
1202 #main start
1203 w()
1204
```

Uninstallation Program(Codes):-

```
*uninstallation hospital.py - /Users/mac/Downloads/HMS Apoorva, Aayushi and Anubhav 12th Science/uninstallation hospital.py (3.8.5)*
__author__ = "APOORVA JADHAV, AAYUSHI KAPOOR & ANUBHAV UTKARSH"
__copyright__ = "Copyright (C) 2021 AAA"
__license__ = "Public Domain"
__version__ = "1.0"

import mysql.connector as mc
from mysql.connector import Error
import tkinter as tk
import tkinter.messagebox as tmb
from tk import *

def uninstall():
    global E, window
    p = E.get()

    try:
        conn_ob=mc.connect(host="localhost",user="root",password=p)

        if (conn_ob.is_connected()):
            q="drop database hospital;"
            query_ob=conn_ob.cursor()
            query_ob.execute(q)
            conn_ob.commit()
            conn_ob.close()

        tmb.showinfo("Uninstalling...","UNINSTALLATION COMPLETED !!!")

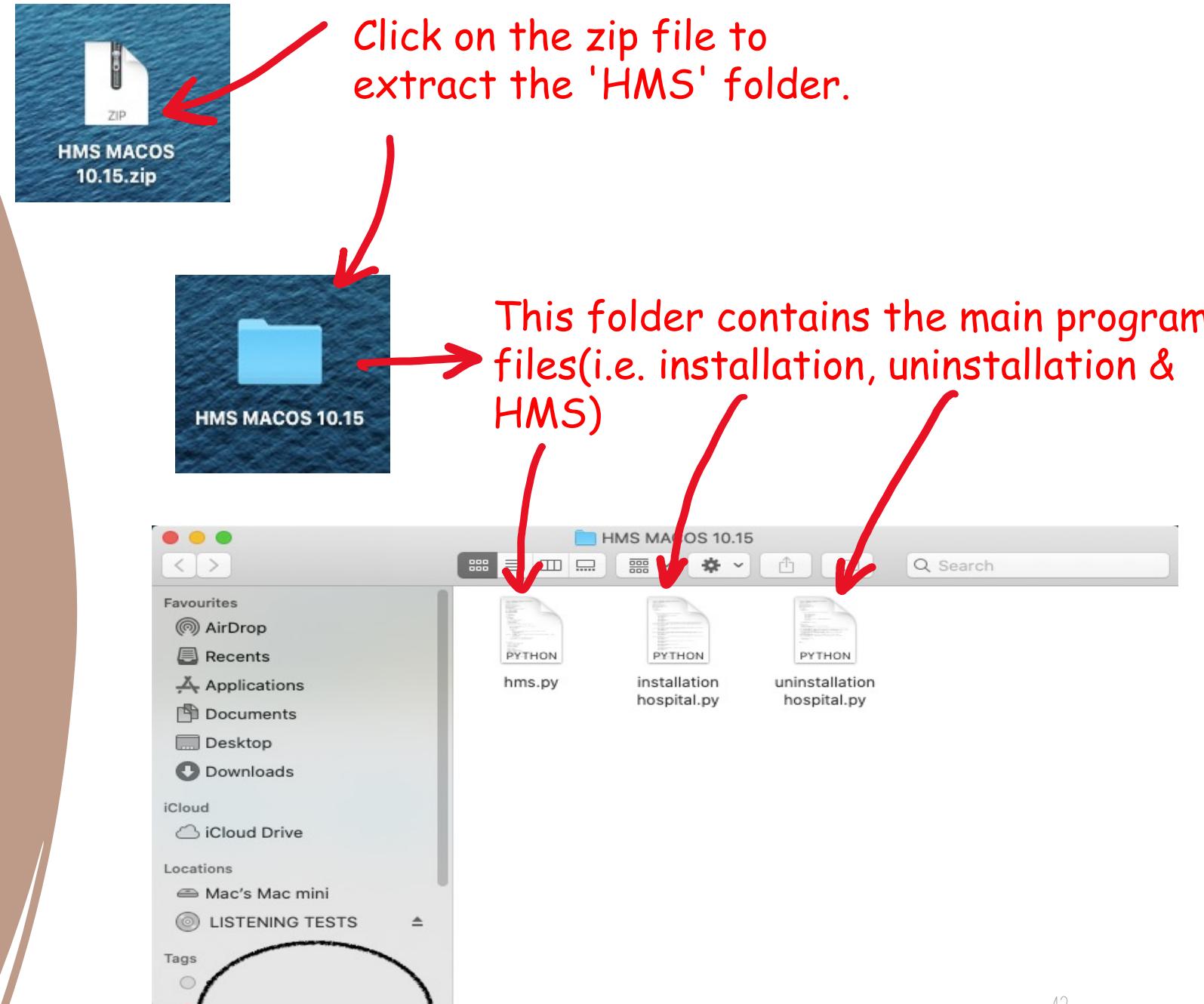
        window.destroy()

    except Error as e:
        tmb.showinfo("Error occured",e)

def w():
    global E,window
    window = tk.Tk()
    window.title("Hospital Management Sysytem Uninstallation Wizard")
    window.geometry("1000x600")
    L1 = tk.Label(window, text = "Hospital Management Sysytem Uninstallation Wizard", font = ("Arial", 20), bg = "pink", fg = "blue").pack(fill="x")
    L2 = tk.Label(window, text = "WARNING", font = ("Arial", 20), fg = "red").pack(pady=20)
    L3 = tk.Label(window, text = "All the data including database and all the tables will be deleted !!!", font = ("Arial", 20), fg = "red").pack(pady=10)
    L4 = tk.Label(window, text = "Enter your MySQL password: ", font = ("Arial", 20)).pack(pady=30)
    E = tk.Entry(window, width = 50, bg="black", fg="white", show="*")
    E.pack(pady = 20)
    B2 = tk.Button(window, text = "Exit", font = ("ComicSansMS", 20), fg = "green", command = window.destroy).place(x=0,y=500)
    B1 = tk.Button(window, text = "UNINSTALL NOW", font = ("ComicSansMS", 20), fg = "red", command = uninstall).place(x=805,y=500)
    window.resizable(False, False)
    window.mainloop()

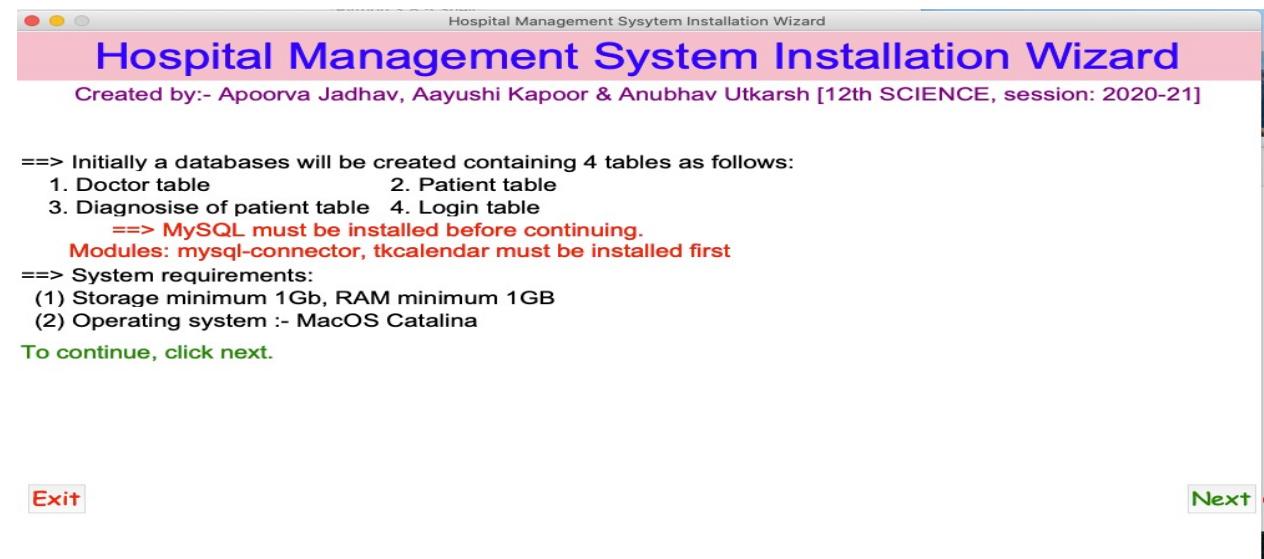
#main start
w()
```

Installation Process



Installation Process

On clicking the installation file the installation program will run and a window will open specifying the information about the tables databases which will be created.

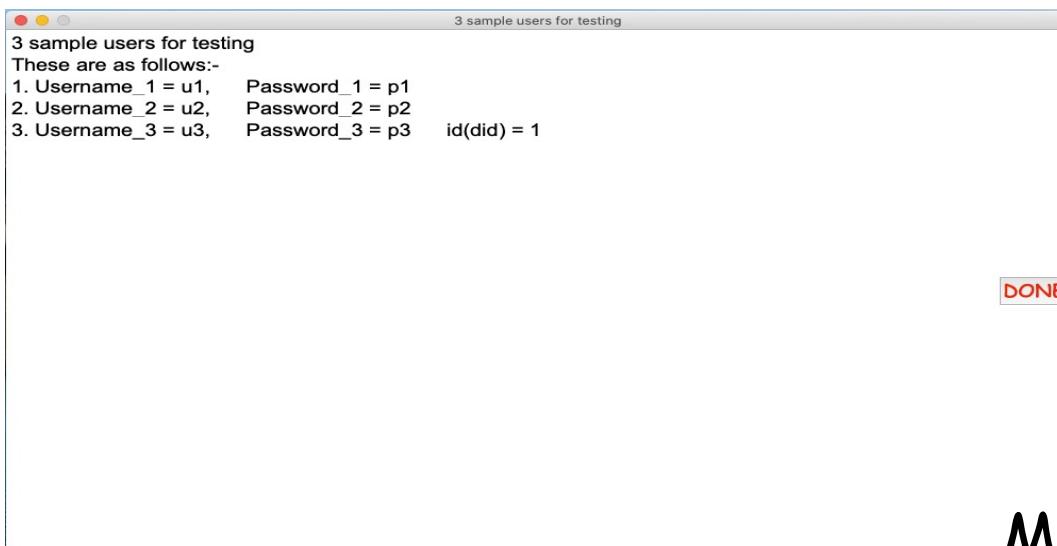


Clicking 'Next' will open another window Asking the user to enter his/her password.



If the user enters the wrong password a message box will appear showing 'Access Denied'.

After entering the correct password the user will be send to a new window displaying the content created by installing the program.



On clicking 'DONE' the window will close and the program will finally be installed.

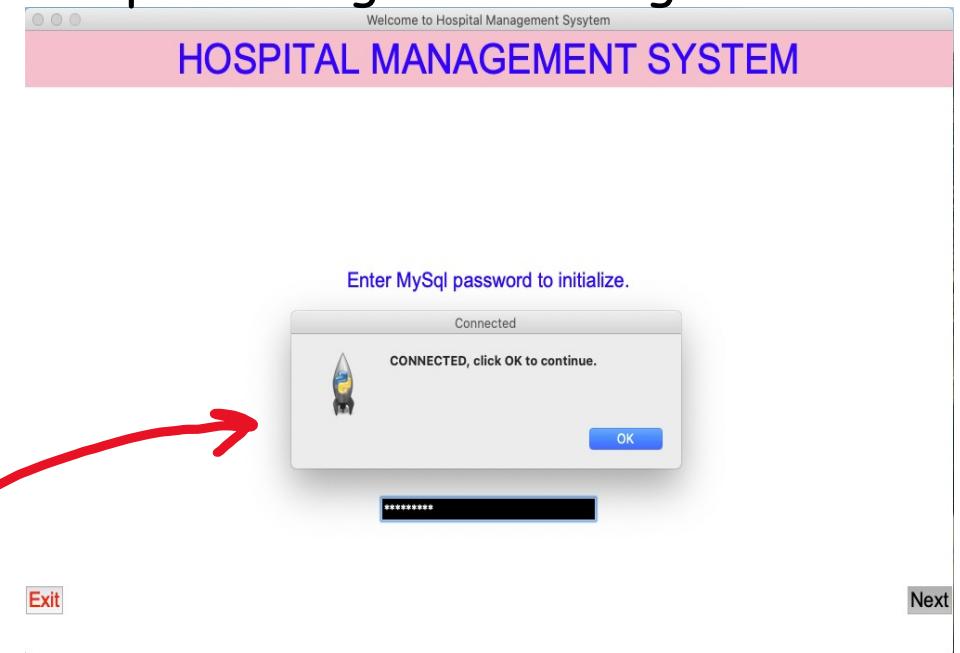
MySql Results

```
mysql> show tables;
+-----+
| Tables_in_hospital |
+-----+
| d_patient           |
| doctor              |
| login               |
| patient             |
+-----+
4 rows in set (0.00 sec)
```

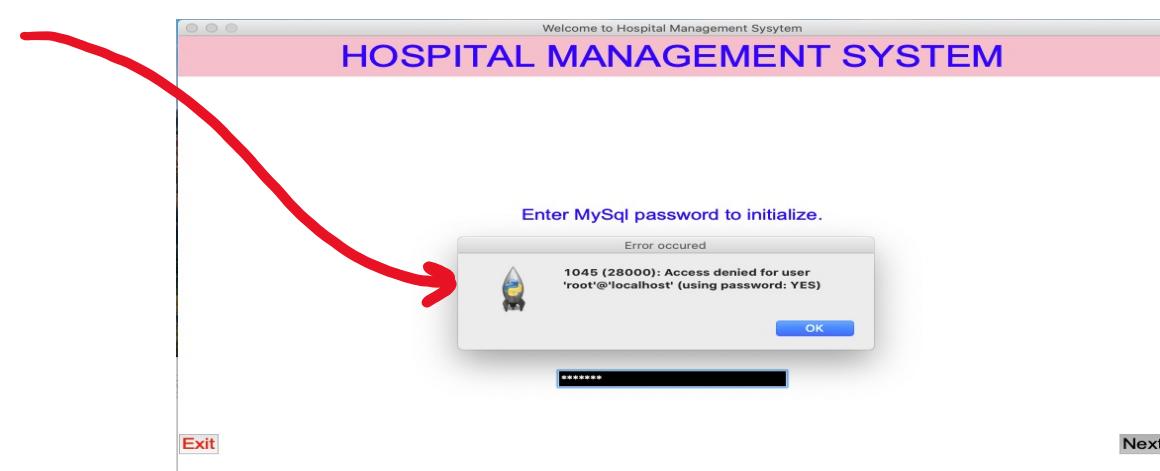
```
mysql> select * from login;
+-----+-----+-----+-----+
| username | password | role | id   |
+-----+-----+-----+-----+
| u1      | p1     | 1    | NULL |
| u2      | p2     | 2    | NULL |
| u3      | p3     | 3    | 1    |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Hospital Management System (Main Program)

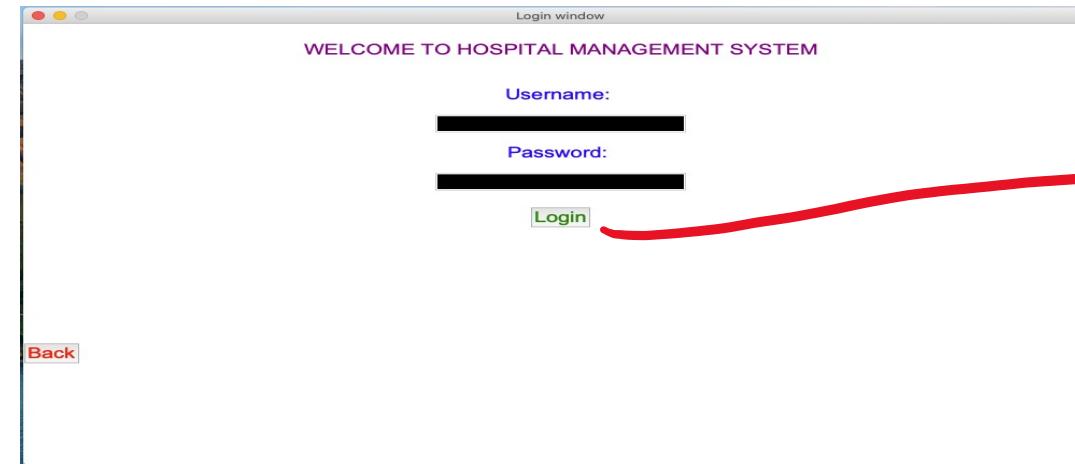
On clicking the 'HMS' file from the main folder a window will open asking the user again for his/her MySql password for initializing the program.



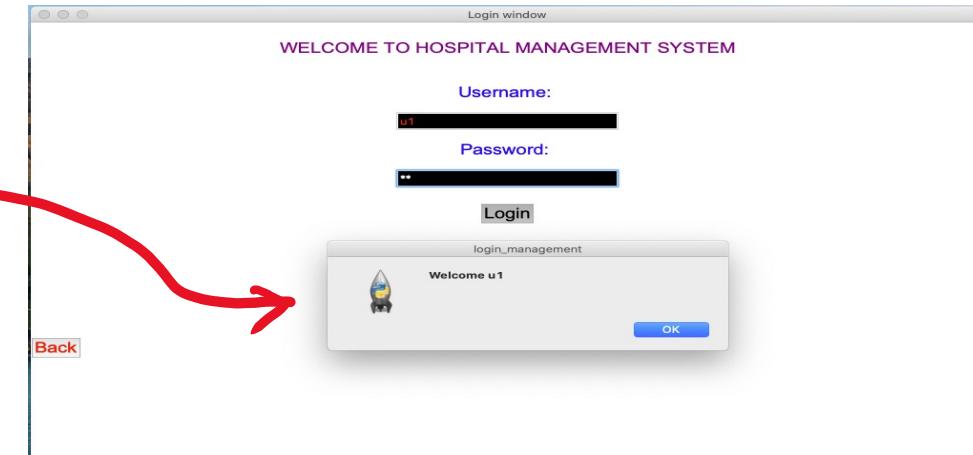
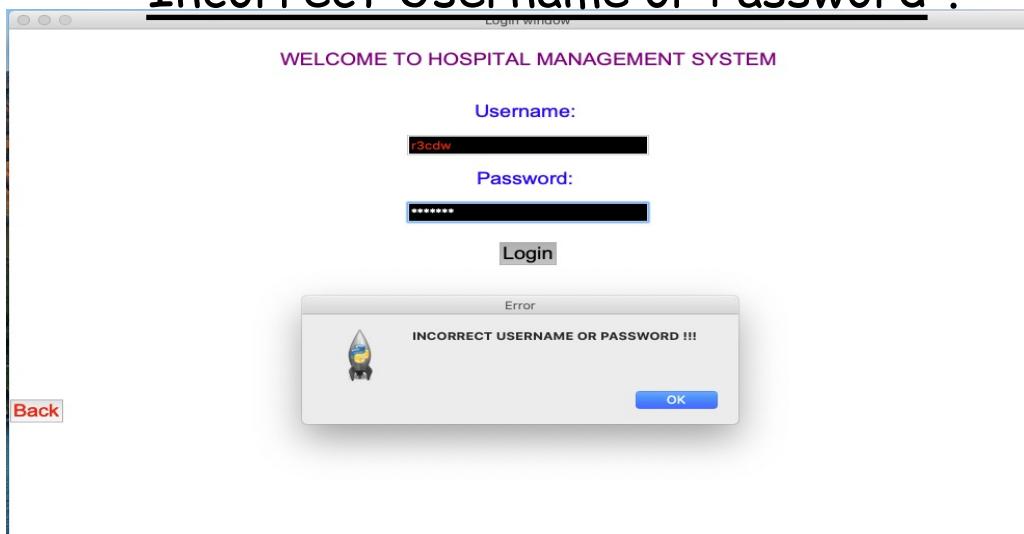
If user enters the wrong password.



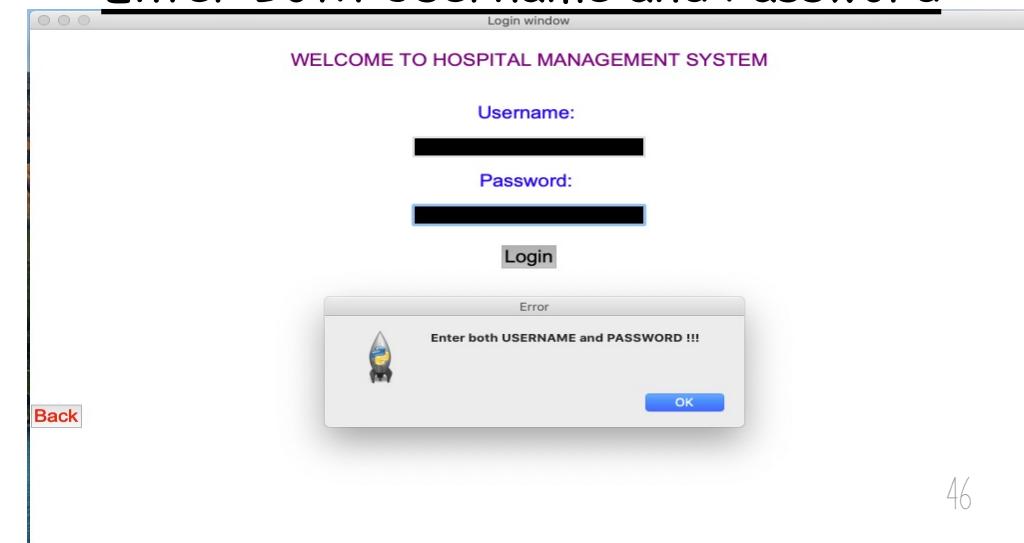
After correctly entering MySql password the user will be taken to a window displaying '**WELCOME TO HOSPITAL MANAGEMENT SYSTEM**', where the user has to enter the username and password as per his chosen work among the 3 users (u1: login management, u2: Receptionist, u3: Doctor).



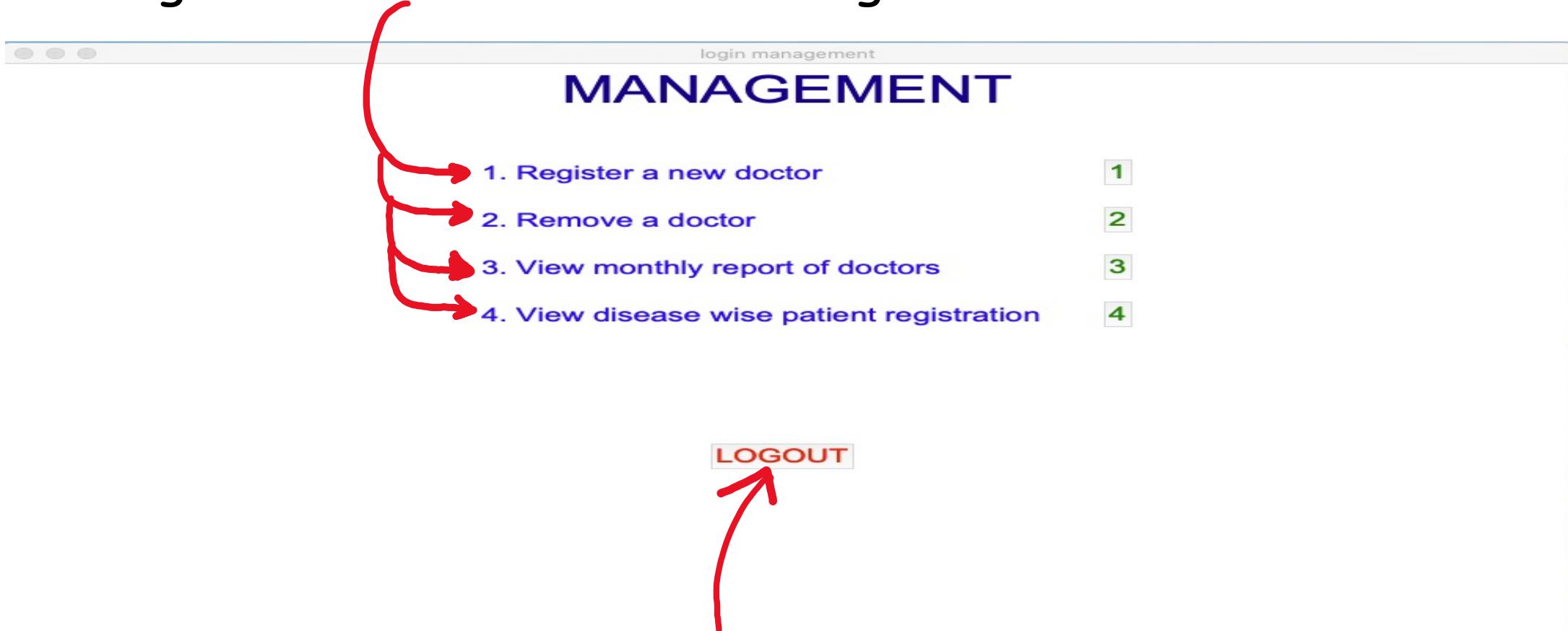
If wrong password entered, the message box will display:
'Incorrect Username or Password'.



If no password or username has been entered by the user then message box will display:
'Enter Both Username and Password'



If the user logins as u1(i.e. login management), a window will open showing the **functions** of the management



On clicking '**LOGOUT**' the program will take the user to the login page again.

Management Work 1(Registration of a new doctor) -

For registering a new doctor the user will click '1' on the management window which will take the user to a window taking the information of the doctor as entries

create doctor id
REGISTER DOCTOR

**Last did value:1

Create doctor's id: []

Enter the name of the doctor: []

Enter the speciality of the doctor: []

Enter the gender of doctor: Gender []

Exit Next

If the leave all the entries and tries going further a message box will pop and show 'Enter every field correctly'.



create doctor id
REGISTER DOCTOR

**Last did value:1

Create doctor's id: 2 []

Enter the name of the doctor: Apoorva []

Enter the speciality of the doctor: Corona []

Enter the gender of doctor: M
Gender
M
F
O

Exit Next

If all the entries are filled correctly, clicking next will lead the user to a new window to create a doctor 'username' and 'password'.

create doctor userpass

MANAGEMENT

~~~ DO NOT CLOSE THE CURRENT WINDOW ~~~

Create username:

Create password:

Re-enter password:

**Next**

create doctor userpass

**MANAGEMENT**

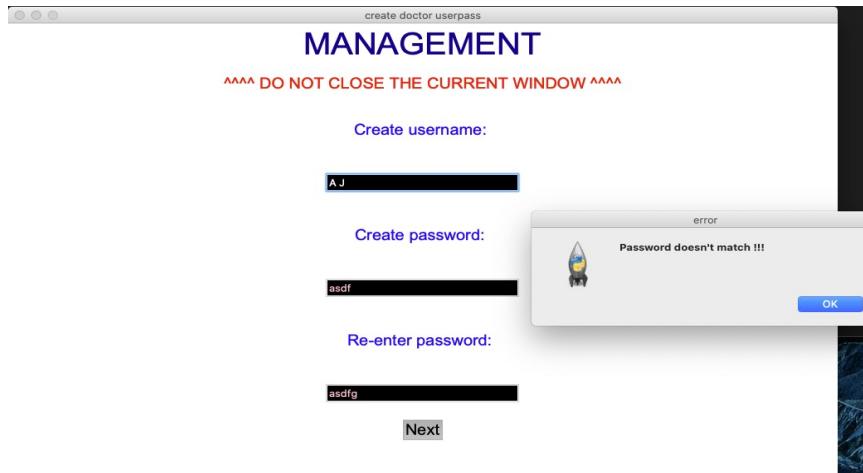
~~~ DO NOT CLOSE THE CURRENT WINDOW ~~~

Create username:
 AJ

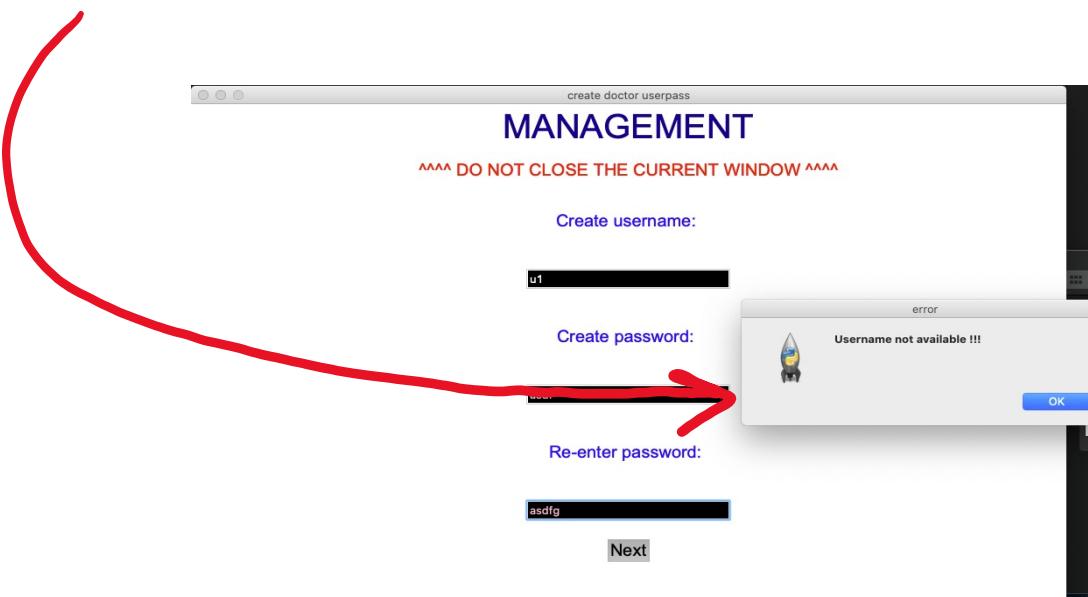
Create password:
 asdf

Re-enter password:
 asdfg

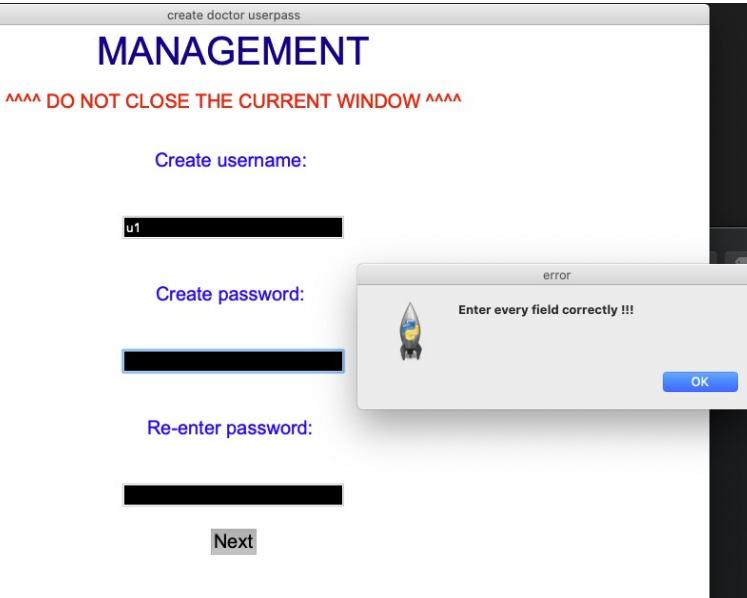
Next



If the user enter an username which already exists, then the message box will display 'Username not available'.



If the user writes different password in the password confirmation entry then the box will show 'Password doesn't match'.



If the user leaves all/some entries then the message box will show 'Enter every field correctly'.

MySql Results

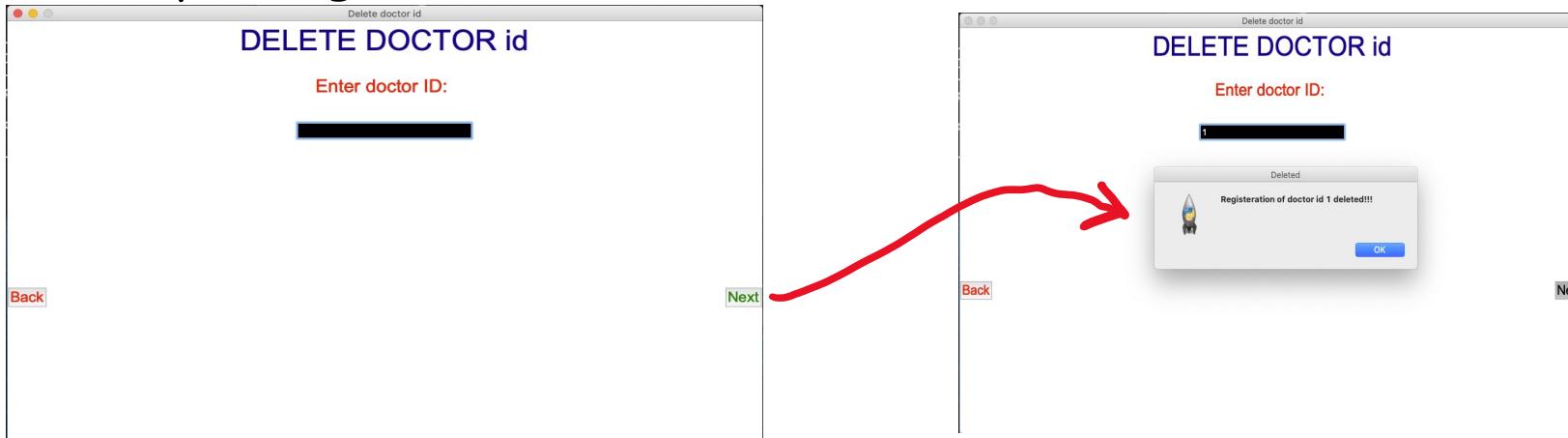
```
mysql> select * from doctor;
+---+-----+-----+-----+
| did | dname | speciality | gender |
+---+-----+-----+-----+
| 1 | Test doctor | Test Speciality | 0 |
| 2 | Apoorva | Corona | M |
| 3 | Aayushi | Food poison | F |
| 4 | Anubhav | Cancer | M |
+---+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
+-----+-----+-----+
| did | dname | speciality | gender |
+-----+-----+-----+
| 1 | Test doctor | Test Speciality | 0 |
| 2 | Apoorva | Corona | M |
+-----+-----+-----+
```

```
mysql> select * from login;
+-----+-----+-----+-----+
| username | password | role | id |
+-----+-----+-----+-----+
| A J | asdf | 3 | 2 |
| A K | 1234 | 3 | 3 |
| A U | 4321 | 3 | 4 |
| u1 | p1 | 1 | NULL |
| u2 | p2 | 2 | NULL |
| u3 | p3 | 3 | 1 |
+-----+-----+-----+
6 rows in set (0.00 sec)
```

Management Work 2(Removal of a doctor) -

For removal of a new doctor the user will click '2' on the management window which will take the user to a window where the user has to enter the doctor id to delete the doctor name corresponding to the id from the table.



MySQL Results

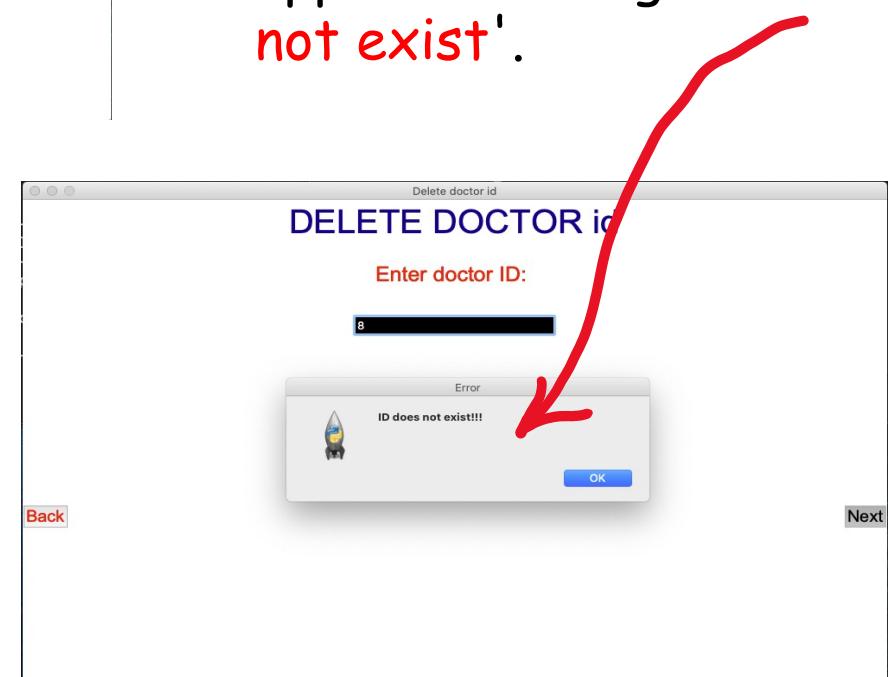
Before removal:

```
mysql> select * from login;
+-----+-----+-----+-----+
| username | password | role | id |
+-----+-----+-----+-----+
| A J     | asdf    | 3    | 2   |
| A K     | 1234    | 3    | 3   |
| A U     | 4321    | 3    | 4   |
| u1      | p1      | 1    | NULL |
| u2      | p2      | 2    | NULL |
| u3      | p3      | 3    | 1   |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

After removal:

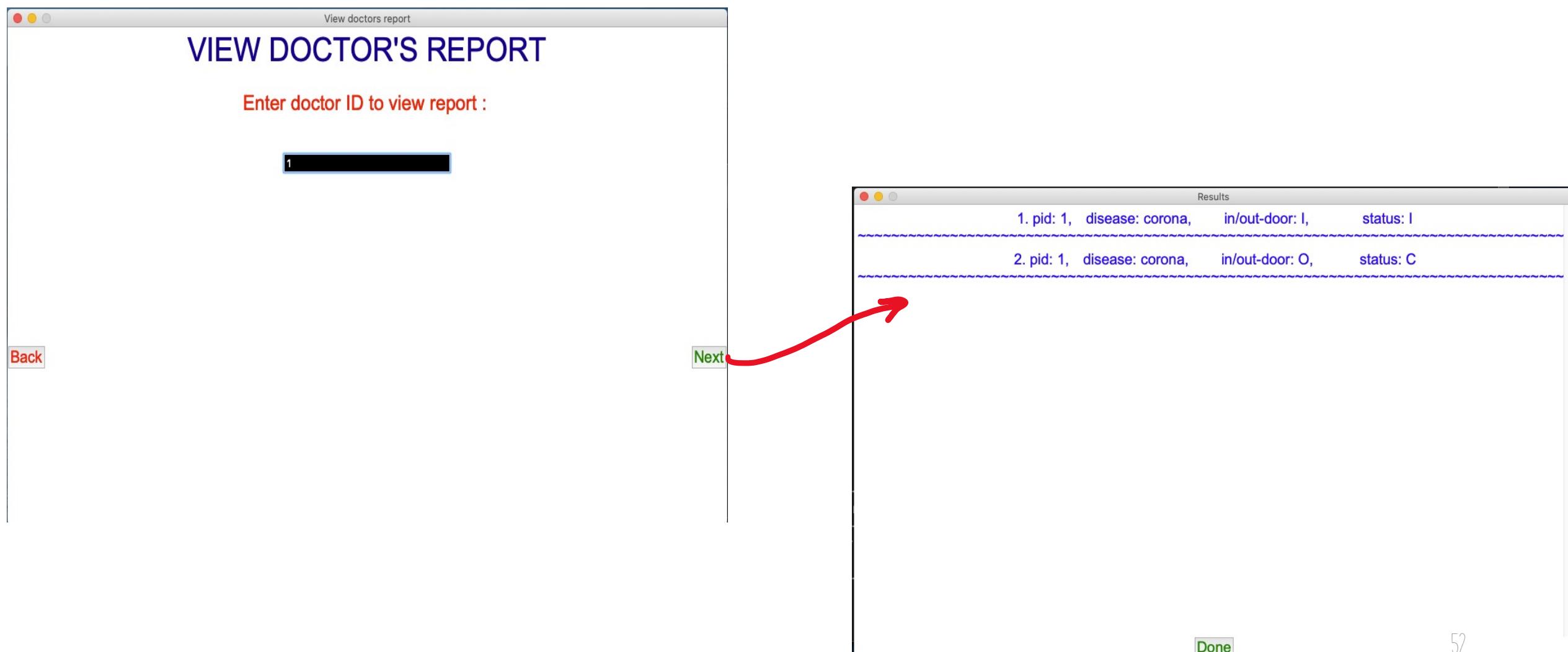
```
mysql> select * from login;
+-----+-----+-----+-----+
| username | password | role | id |
+-----+-----+-----+-----+
| A J     | asdf    | 3    | 2   |
| A K     | 1234    | 3    | 3   |
| A U     | 4321    | 3    | 4   |
| u1      | p1      | 1    | NULL |
| u2      | p2      | 2    | NULL |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

If the management enters the wrong id that do not exists then a message will appear showing ' ID does not exist'.



Management Work 3(Monthly report of doctor) -

For viewing the monthly report the user will click '3' on the management window which will take the user to a window where the user will enter the valid doctor ID to check the monthly report of the doctor.



Management Work 4 (View disease wise patient registration) -

For viewing disease wise patient report, the user will click on '4' on Management window which takes the users to window where he/she will have to enter disease and its status.

view patient report

VIEW PATIENT REPORT

Enter patient's disease:

corona

Enter patient's status (C:cured, I:Infected):

I

Back

Next



Patient's report

1. pid: 1 , name: Abhinav, address: Patna

Done

view patient report

VIEW PATIENT REPORT

Enter patient's disease:

corona

Enter patient's status (C:cured, I:Infected):

c

Back

Next



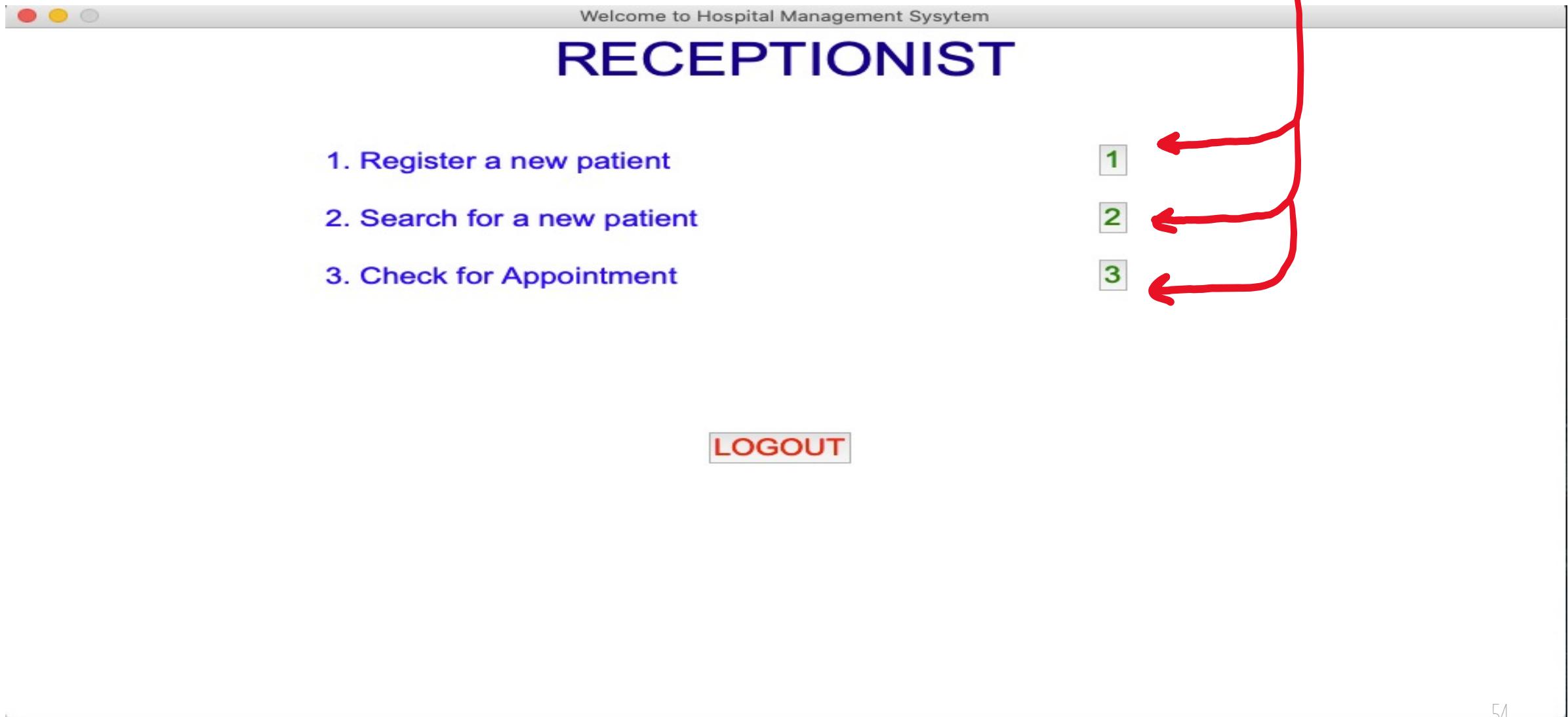
Patient's report

1. pid: 1 , name: Abhinav, address: Patna

Done

Receptionist

If the user logins as u2(i.e. Receptionist), a window will open showing the **functions** of the Receptionist



Receptionist Work 1(Registration of patient):-

For registering a new patient the user will click '1' on the receptionist window which will take the user to a window taking the information of patient as entries.

reg new pat

REGISTER A NEW PATIENT

**Last pid value:4

Create patient ID :

Enter patient name :

Enter patient's gender : Choose Gender

Enter Blood Group : Choose Blood Group

Enter patient's city :

Enter patient's age : Choose age

reg new pat

REGISTER A NEW PATIENT

**Last pid value: 0

Create patient ID :

Enter patient name :

Enter patient's gender : M

Enter Blood Group : B+ve

Enter patient's city : Patna

created
Patient id 1 registered successfully.

If the user registers the patient with an ID that already exists for another patient then the message box will show:

'Patient ID already exists'

reg new pat

REGISTER A NEW PATIENT

**Last pid value:1

Create patient ID :

Enter patient name : KK

Enter patient's gender : O

Enter Blood Group : O+ve

Enter patient's city : Delhi

Error
Patient ID already exists!!!
Change patient id to continue.

If the user registers every field correctly.

MySql Results:-

```
mysql> show tables;
+-----+
| Tables_in_hospital |
+-----+
| d_pat_1              |
| d_patient            |
| doctor               |
| login                |
| patient              |
+-----+
5 rows in set (0.00 sec)
```

```
mysql> select * from patient;
+-----+-----+-----+-----+-----+
| pid | pname | gender | address | blood_group | reg_pat_date |
+-----+-----+-----+-----+-----+
|   1 | Abhinav | M | Patna | B+ve | 2021-01-24 12:09:19 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Receptionist Work 2(Searching data of a patient):-

The screenshot shows a "SEARCH A PATIENT" window with the following fields:

- Enter patient's name:
- Enter patient's blood group:
Choose Gender
M (selected)
F
O

A red arrow points from the "Search" button to the right window.

The right window displays the search results:

Result

1. Patient's id (PID): 3

Done

The screenshot shows a "SEARCH A PATIENT" window with the following fields:

- Enter patient's name:
- Enter patient's blood group:
Choose Gender
M
F (selected)
O

A red arrow points from the "Search" button to the right window.

The right window displays the search results:

Result

1. Patient's id (PID): 2

2. Patient's id (PID): 4

Done

Receptionist Work 3(Checking for appointment):-

check appoinment

SEARCH FOR APPOINTMENT

Enter patient's id (PID no):

1

Back

Check

check appoinment

SEARCH FOR APPOINTMENT

Enter patient's id (PID no):

1

result

Pid: 1
Last Appointment date: 2021-01-26

OK

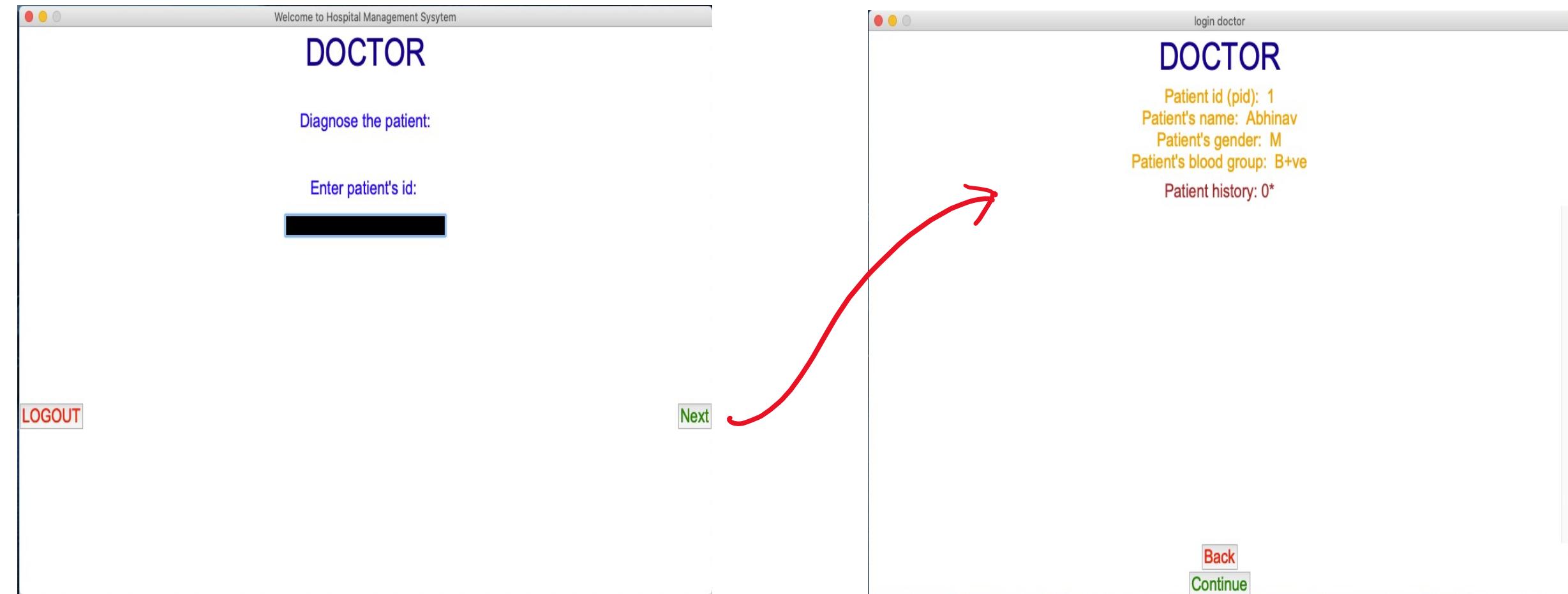
Back

Check

A red curved arrow starts at the 'Check' button on the left screen and points to the 'result' dialog on the right screen.

Doctor:-

If the user logs in as u3 (i.e. Doctor), a window will open showing the user to enter the patient ID for the diagnose of that patient.



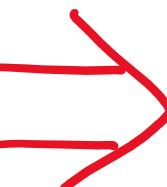
Making report of the patient:-

diagnosis patient

DOCTOR

1. Enter symptoms of patient:
2. Enter disease of patient:
3. Enter diagnosis done or suggested:
4. Enter medicines prescribed:
5. Enter whether patient is indoor or outdoor [I,O]: Select choice
6. Enter ADMIT date (YYYY-MM-DD): Select date
7. Enter DISCHARGE date (YYYY-MM-DD): Select date
8. Enter next APPOINTMENT date (YYYY-MM-DD): Select date
9. Cured-'C'/Infected-'I': Select choice

Continue



diagnosis patient

DOCTOR

1. Enter symptoms of patient: fever
2. Enter disease of patient: corona
3. Enter diagnosis done or suggested: corona test
4. Enter medicines prescribed: pfizer
5. Enter whether patient is indoor or outdoor [I,O]: Select choice
6. Enter ADMIT date (YYYY-MM-DD): Select date 2021-01-24
7. Enter DISCHARGE date (YYYY-MM-DD): Select date None
8. Enter next APPOINTMENT date (YYYY-MM-DD): Select date
9. Cured-'C'/Infected-'I': Select choice

Continue

diagnosis patient

January 2021

| Mon | Tue | Wed | Thu | Fri | Sat | Sun |
|-----|-----|-----|-----|-----|-----|-----|
| 53 | 28 | 29 | 30 | 31 | 1 | 2 |
| 1 | 4 | 5 | 6 | 7 | 8 | 9 |
| 2 | 11 | 12 | 13 | 14 | 15 | 16 |
| 3 | 18 | 19 | 20 | 21 | 22 | 23 |
| 4 | 25 | 26 | 27 | 28 | 29 | 30 |
| 5 | 1 | 2 | 3 | 4 | 5 | 6 |

None Ok



diagnosis patient

DOCTOR

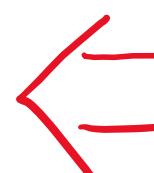
1. Enter symptoms of patient: none
2. Enter disease of patient: corona
3. Enter diagnosis done or suggested: corona test
4. Enter medicines prescribed: none
5. Enter whether patient is indoor or outdoor [I,O]: O Select choice
6. Enter ADMIT date (YYYY-MM-DD): Select date None
7. Enter DISCHARGE date (YYYY-MM-DD): Select date 2021-01-24
8. Enter next APPOINTMENT date (YYYY-MM-DD): Select date None
9. Cured-'C'/Infected-'I': Select choice

Continue

Done

Diagnose recorded !!!

OK



diagnosis patient

DOCTOR

1. Enter symptoms of patient: none
2. Enter disease of patient: corona
3. Enter diagnosis done or suggested: corona test
4. Enter medicines prescribed: none
5. Enter whether patient is indoor or outdoor [I,O]: O Select choice
6. Enter ADMIT date (YYYY-MM-DD): Select date None
7. Enter DISCHARGE date (YYYY-MM-DD): Select date 2021-01-24
8. Enter next APPOINTMENT date (YYYY-MM-DD): Select date None
9. Cured-'C'/Infected-'I': Select choice

Continue

Viewing doctors report:-

View doctors report

VIEW DOCTOR'S REPORT

Enter doctor ID to view report :

1

[Back](#)

[Next](#)

login doctor

DOCTOR

Patient id (pid): 1
Patient's name: Abhinav
Patient's gender: M
Patient's blood group: B+ve
Patient history: 1*

History: 1 ,
Entry date: 2021-01-24 13:49:24 ,
Symptoms: fever ,
Disease: corona ,
Diagnosis: corona test ,
Medicines given: pfizer ,
Patient admitted or not: || ,
Admit date (if admitted): 2021-01-24 00:00:00 ,
Discharge date: None ,
Appointment date: 2021-01-26 00:00:00 ,

[Back](#)

[Continue](#)

MySql Results:-

1.

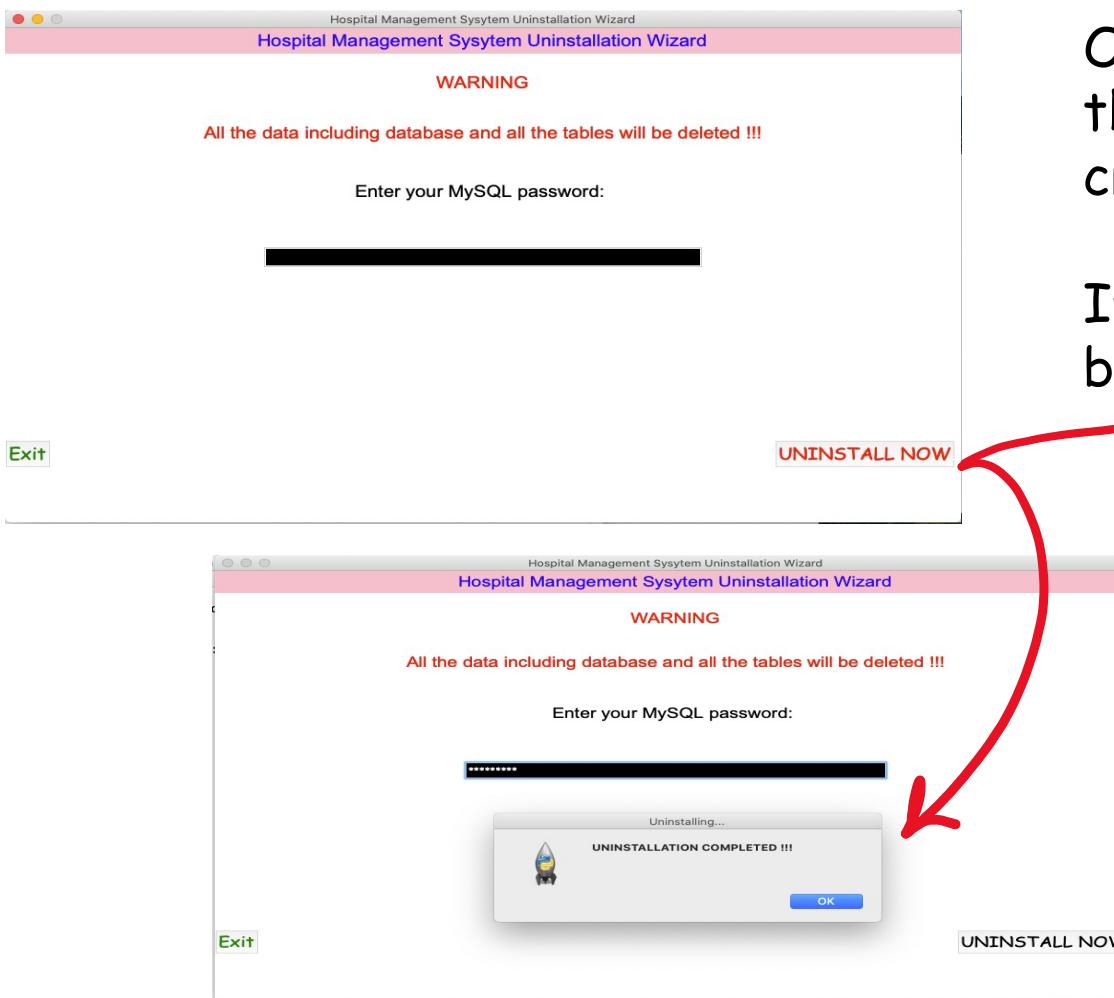
```
[mysql] > select * from d_pat_1;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| entry_dates | symptoms | disease | diagnosis | medicines | iop | admit_date | discharge_date | appointment_date |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 2021-01-24 13:49:24 | fever | corona | corona test | pfizer | I | 2021-01-24 00:00:00 | NULL | 2021-01-26 00:00:00 |
| 2021-01-24 13:51:46 | none | corona | corona test | none | 0 | NULL | 2021-01-24 00:00:00 | 1000-01-01 00:00:00 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

2.

```
[mysql] > select * from d_patient;
+-----+-----+-----+-----+-----+
| pid | did | disease | iop | status |
+-----+-----+-----+-----+-----+
| 1 | 1 | corona | I | I |
| 1 | 1 | corona | O | C |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Uninstallation Process

To uninstall the program click on the uninstallation file from the main folder. This will lead the user to a window which will ask the user to enter his/her MySql password for the unistallation process.



On clicking 'UNINSTALL' the user will uninstall the program and all the tables and the database created while installation will be deleted.

OR

If the user enters the wrong password a message box will pop-up and show 'Access Denied'.

