**PROJECT TITLE: IOT BASED HEALTH MONITORING SYSTEM**

**PROJECT LEADER: SHAMBHAVI TOLNUR(2BL15CS092)**

**Department : Computer Science & Engineering**

**BLDEA’s Dr. P.G.Halakatti College of Engineering & Technology, Vijayapur-586103.**

**Installation and Setup**

#1 Raspberry pi setup

Step 1:Insert the memory card in the Raspberry pi board.

Step 2:Load the Rasbian os into the memory card.

Step 3:Login to Raspbian os through Wifi using ssh pi@IPaddress cmd.

Step 4:Stop.

#2 DBMS installation

Step 1: install mysql.

Step 2:Draw the ER diagram of patient and health info.

Step 3:create two tables namely patient\_info and health\_info.

Step 4: normalize the table to 3NF.

Step 5: stop.

#3 Client server interaction

Step 1:Login to server IPaddress through WiFi module.

Step 2: Register the patient details into server’s patient\_info table.

Step 3: Enroll.py

import MySQLdb

import os

import time

#connect to database

try:

db=MySQLdb.connect(“192.168.43.113”, “root”, “password”, “healthCare”)

except Exception as e:

print(‘error opening database!’)

print(‘exception message;’ +str(e))

exit(1)

##gets some sensor information

while True:

print(‘enter patient information\n’)

name=raw\_input(‘enter name of the patient:’)

age=raw\_input(‘enter age of the patient:’)

gender=raw\_input(‘enter name of the patient:’)

gender=raw\_input(‘enter name of the patient:’)

address=raw\_input(‘enter address of the patient:’)

city=raw\_input(‘enter city of the patient:’)

phone=raw\_input(‘enter name of the patient:’)

#add to db

cur=db.cursor()

try:

stmt = ‘insert into patient(name,age,gender,address,city,phone,date,time) values(“%s”, %d, “%s”, “%s”,“%s”,“%s”,CURDATE(),CURTIME())’%(name,age,gender,address,city,phone)

cur.execute(stmt)

db.commit()

except:

db.rollback()

conti = raw\_input(‘do you wish to add one more entry ?(yes/no) :’)

if conti!=’yes’:

break

Step 4: Stop

.

#4 Sensors setup

Step 1:Temperature sensor(LM35) has three pins namely Vcc,ground and

Signal.

Step 2: Connect the pins of LM35 to the Raspberry pi. Connect Vcc-pin 1,

ground-pin 6 and signal-pin 7 through jumper wires.

Step 3: stop.

#5 Reading data from sensors

Step 1: Reading LM35.py

#!/usr/bin/env python

from pulsesensor import Pulsesensor

import MySQLdb

import os

import time

import glob

def heart():

p = Pulsesensor()

p.startAsuncBPM()

try:

while True:

bpm = p.BPM

if bpm > 55 and bpm<72:

return bpm

else:

pass

time.sleep(1)

except:

p.stopAsyncBPM()

#initialize the GPIO pins

os.system(‘modprobe w1.gpio’) # turns on the GPIO module

os.system(‘modprobe w1.thern’) # turns the temperature module

#finds the correct device file that holds the temperature data

base\_dir = ‘/sys/bus/w1/devices/’

device\_folder = glob.glob(base\_dir + ‘28\*’)[0]

device\_file = device\_folder + ‘/w1\_slave’

# A function that reads the sensors data

def read\_temp\_raw():

f= open(device\_file, ‘r’) # opens the temperature device file

lines = f.readlines() # returns the text

f.close()

return lines

# convert the value of the sensor into a temperature

def read\_temp:

lines = read\_temp\_raw() # read the temperature ‘device file’

# while the first line does not contain ‘YES’, wait for 0.2s

#and then read the device file again

while lines[0].strip()[-3:]!= ’YES’:

time.sleep(0,2)

lines = read\_temp\_raw()

# look for the position of the ‘=’ in the second line of the device file

equals\_pos = lines[1].find(‘t=’)

# if the ‘=’ is found, convert the rest of the line after the ‘=’ into degree clesius ,

# then degrees farenheit

if equals\_pos !=-1:

temp\_string = lines[1][equals\_pos+2:]

temp\_c = float(temp\_strig)/1000.0

return temp\_c

# connect to database

try :

db=MySQLdb.connect(“192.168.43.113”, “root”, “password”, “healthcare”) #put IP address of # the server

except Exception as e:

print(‘error opening database!’)

print(‘exception message: ‘ +str(e))

exit(1)

## gets some sensor information

while True:

pid= int(input((‘enter patient id(pid):’)))

print(‘connect all the sensors to the patient\n’)

procced=raw\_input(‘have you connected all the sensors(Y/N): ‘)

if proceed==’Y’:

temp\_c=read\_temp()

temp\_f=temp\_c \* 9.0 / 5.0 + 32.0

print(temp\_c, temp\_f)

time.sleep(1)

# heart\_rt=heart();

#add to db

cur=db.cursor()

try:

stmt = ‘insert into health\_info(pid,temperature,time) values(%d,%d,CURTIME())’ %(pid,int(temp\_c))

cur.execute(stmt)

db.commit()

except:

db.rollback()

conti = raw\_input(‘do you wish to add one more entry ?(yes/no) :’)

if conti!=’yes’:

break

Step 2: Temperature of patient is displayed in degree ℃ and ℉ on LCD display.

Step 3: The displayed value is stored in health\_info table of server.

Step 4: stop.

**Note**: **The module #4 and #5 are repeated for other sensors like heart rate,ECG and blood**

**Pressure.**

#6 Testing

Step 1: All the sensors are connected to the patient.

Step 2:Different sensor parameters( Temperature, heart rate,ECG and blood Pressure) are

recoreded and stored into the health\_info table.

Step 3:The database contents are analysed by the doctor for patient disease diagnosis.