Certification Page

This page must be the first page of your uploaded document.
Your assignment will not be graded without this page (completed with your full name in the area provided) as the first page of your uploaded document.

I, ___***Ulvi Bajarani***___________________, certify that the work I am uploading represents efforts of my team member and mine per the following task-contribution table, and is not copied from anyone else or any other resource (such as Internet).

	Tasks on the schedule for this quiz are 1, 2, 3, and 4. You may break these tasks further down into subtasks as needed.	Percent contributed	Percent contributed	Total percent
#	*Subtask* Description	Ulvi Bajarani	Gwendolyn Poulos	
1.1	Program Android App to interact with DB Tables in PaaS	50%	50%	100%
1.2	Test interaction between Android App and RPi through DB tables in PaaS	50%	50%	100%
1.3	Fixing Chart on the website	50%	50%	100%

The updated code of welcome.php (includes the link to the chart)

```
<?php
require_once __DIR__ . '/../required/db_connect.php';
?>
<html>
  <header>
    <title>Welcome</title>
    <meta charset="UTF-8"/>
  </header>
  <body>
    <div id="header">
      <h1>Welcome</h1>
    </div>
    <div id="body">
      <div id="status"></div>
      <script type="text/javascript" src="jquery.js"></script>
      <script src="https://cdn.jsdelivr.net/npm/chart.js@2.8.0"></script>
      <script type="text/javascript">
        $(document).ready(function() {
          setInterval(function() {
            $('#status').load('DBdevice.php')
          }, 3000);
        });
      </script>
      <form id="form1" method="post" action="">
        <input type="hidden" id="out" value="1">
        <button type="submit">Logout</button>
      </form>
      <div id=check>
        <?php
          if(isset($_POST['out'])){
          $uname='ben';
          $id='10';
          date_default_timezone_set("America/Chicago");
          $date=date("Y-m-d H:i:s");
          $stmt=$mysqli->prepare("INSERT INTO
TransactionalLogs(TimestampInfo,MessageID,DataInformation) VALUES(?,?,?)");
          $stmt->bind_param('sss',$date,$id,$uname);
          $stmt->execute();
          $stmt->close();
          echo "<script type='text/javascript'>alert('Logging Out')</script>";
          echo "<script> window.location.assign('index.php');</script>";
          }
        ?>
      </div>
      <div id="chartLink"><a href="Chart.php">The link to the Timestamp
chart</div>
    </div>
  </body>
</html>
```

The code of data.php (to fetch the data for the chart)

```
<?php
require_once __DIR__ . '/../required/db_connect.php';

$stmt = "SELECT MessageID, TimestampInfo FROM TransactionalLogs ORDER BY
TimestampInfo";
$result = mysqli_query($mysqli,$stmt);
$data = array();
foreach ($result as $row){
  $data[] = $row;
}
mysqli_close($mysqli);
echo json_encode($data);
?>
```

The code of Chart.php (The chart code)

```
<html>
  <!--https://phppot.com/php/creating-dynamic-data-graph-using-php-and-chart-js/-->
  <div id="charts">
    <canvas id="chart1"></canvas>
  </div>
  <script src='https://cdn.jsdelivr.net/npm/chart.js@2.8.0'></script>
  <script src='jquery.js'></script>
  <script>
    $(document).ready(function (){
      showGraph();
    });
```

```
function showGraph(){
    {
        $.post('data.php',
        function(data){
            console.log(data);
            data=JSON.parse(data);
            var time=[];
            var mid=[];

            for (var i in data){
                time.push(data[i].TimestampInfo);
                mid.push(data[i].MessageID);
            }

            var chartdata={
                labels: time,
                datasets:[{
                    label: 'LED Monitoring',
                    borderColor: '#F778A1',
                    data: mid
                }]
            };
            var graphTarget = $("#chart1");

            var linegraph = new Chart (graphTarget,{
                type: 'bar',
                data: chartdata,
                options:{
                    scales:{
                        xAxes:[{
                            scaleLabel:{
                                display: true,
                                fontColor:'#6C2DC7',
                                labelString: 'Time Stamp'
                            },
                            ticks: {
                                fontColor:'#6C2DC7'
                            }
                        }],
                        yAxes:[{
                            scaleLabel:{
                                display: true,
                                        fontColor:'#6C2DC7',
                                        labelString: 'MessageID'
                                    },
                                    ticks: {
                                        fontColor:'#6C2DC7',
                                        suggestedMin:12,
                                        suggestedMax:15
                                    }
                                }]
                            }
                        }
                    });
                });

            }
        }
    </script>
</html>
```

The code of Projectv4.py, which is launched from an Android Phone to test.

```
import time
import kivy
from kivy.app import App
from kivy.uix.button import Button
from kivy.uix.boxlayout import BoxLayout
from kivy.uix.label import Label
from kivy.uix.switch import Switch
from kivy.clock import Clock
from functools import partial
import requests

class BoxLayoutApp(App):

    #def __init__(self, **kwargs):
        #super(BoxLayoutApp,self).__init__(**kwargs)
    def build(self):
        self.superBox = BoxLayout(orientation ='vertical')

        self.VB1 = BoxLayout(orientation ='vertical')

        self.lbl1 = Label(text="Monitoring System")
self.VB1.add_widget(self.lbl1)

localtime = time.asctime(time.localtime(time.time()))
self.lbl2 = Label(text=localtime)
self.VB1.add_widget(self.lbl2)

self.HB1 = BoxLayout(orientation ='horizontal')

self.lbl3 = Label(text="Diastolic BP < 60mmHg:")
self.HB1.add_widget(self.lbl3)

self.sw1=Switch(active=False,
            disabled=True)
self.HB1.add_widget(self.sw1)
#self.sw1.bind(active=switch_callback1)

self.HB2 = BoxLayout(orientation ='horizontal')
self.lbl4 = Label(text="Diastolic BP > 120mmHg:")
self.HB2.add_widget(self.lbl4)

self.sw2=Switch(active=False,
            disabled=True)
self.HB2.add_widget(self.sw2)
#self.sw2.bind(active=switch_callback2)

self.HB3 = BoxLayout(orientation ='horizontal')
self.lbl5 = Label(text="Administering Norepinephrine:")
self.HB3.add_widget(self.lbl5)

self.led1=Switch(active=False,
             disabled=False)
self.HB3.add_widget(self.led1)
self.led1.bind(active=switch_callback3)

self.HB4 = BoxLayout(orientation ='horizontal')
self.lbl6 = Label(text="Administering Nitroglycerin:")
self.HB4.add_widget(self.lbl6)

self.led2=Switch(active=False,
             disabled=False)
self.HB4.add_widget(self.led2)
self.led2.bind(active=switch_callback4)
self.HB5 = BoxLayout(orientation ='horizontal')
self.lbl7 = Label(text="Alarm1")
self.HB5.add_widget(self.lbl7)

self.settings5=Switch(active=False)
self.HB5.add_widget(self.settings5)
self.settings5.bind(active=self.switch_callback5)

self.lbl71=Label(text="On since: ")
self.HB5.add_widget(self.lbl71)
self.btn5 = Button(text="acknowledge")
self.btn5.bind(on_press=self.pressed2)
self.HB5.add_widget(self.btn5)

self.HB6 = BoxLayout(orientation ='horizontal')
self.lbl8 = Label(text="Alarm2")
self.HB6.add_widget(self.lbl8)

self.settings6=Switch(active=False)
self.HB6.add_widget(self.settings6)
self.settings6.bind(active=self.switch_callback6)

self.lbl81=Label(text="On since: ")
self.HB6.add_widget(self.lbl81)

self.btn6 = Button(text="acknowledge")
self.btn6.bind(on_press=self.pressed3)
self.HB6.add_widget(self.btn6)

self.HB7 = BoxLayout(orientation ='horizontal')
self.btn7 = Button(text="Dispense Norepinephrine")
self.btn7.bind(on_press=self.pressed1)
self.HB7.add_widget(self.btn7)
self.btn8 = Button(text="Dispense Nitroglycerin")
self.btn8.bind(on_press=self.pressed)
self.HB7.add_widget(self.btn8)
self.btn9 = Button(text="Dispense Auto")
self.btn9.bind(on_press=self.presseda)
self.HB7.add_widget(self.btn9)

self.superBox.add_widget(self.VB1)
```

```
        self.superBox.add_widget(self.HB1)
        self.superBox.add_widget(self.HB2)
        self.superBox.add_widget(self.HB3)
        self.superBox.add_widget(self.HB4)
        self.superBox.add_widget(self.HB5)
        self.superBox.add_widget(self.HB6)
        self.superBox.add_widget(self.HB7)

        #schedule the JSONrequest function to trigger every 5 seconds to read/write
database
        event = Clock.schedule_interval(partial(self.JSONrequest),5)

        return self.superBox

    def JSONrequest(self, *largs):
        if(self.sw1.active==True):
            SW1=1
        else:
            SW1=0
        if(self.sw2.active==True):
            SW2=1
        else:
            SW2=0
        if(self.led1.active==True):
            LED1=1
        else:
            LED1=0
        if(self.led2.active==True):
            LED2=1
        else:
            LED2=0
        data={'username':'Test',
'password':'Tester3','SW1':SW1,'LED1':LED1,'SW2':SW2,'LED2':LED2}

res=requests.post("https://team2project3342.online/scripts/sync_android_data.php",json
=data)
        print(res)
        r=res.json()
        if(SW1!=r['SW1']):
           print("Changing SW1 status to the value in the database.")
           if self.sw1.active==True:
               self.sw1.active=False
            self.led1.active=False
        else:
            self.sw1.active=True
    if(SW2!=r['SW2']):
        print("Changing SW2 status to the value in the database.")
        if self.sw2.active==True:
            self.sw2.active=False
            self.led2.active=False
        else:
            self.sw2.active=True
    return

def pressed(self,instance):
    print ("you picked " + instance.text)
    self.led2.active=True

def pressed1(self,instance):
    print ("you picked " + instance.text)
    self.led1.active=True

def presseda(self,instance):
    print ("you picked " + instance.text)
    if self.sw1.active==True:
        self.led1.active=True
    else:
        self.led1.active=False
    if self.sw2.active==True:
        self.led2.active=True
    else:
        self.led2.active=False

def pressed2(self,instance):
    print ("you acknowleged Alarm1")
    self.settings5.active=False
    print("turning alarm1 off")
def pressed3(self,instance):
    print ("you acknowleged Alarm2")
    self.settings6.active=False
    print("turning alarm1 off")
def switch_callback5(self, switchObject, switchValue):
    print('Value of Alarm1: ', switchValue)
    if self.settings5.active==True:
            ltm = time.asctime(time.localtime(time.time()))
            self.lbl71.text+=ltm
        else:
            self.lbl71.text = "On since: "
    def switch_callback6(self, switchObject, switchValue):
        print('Value of Alarm2: ', switchValue)
        if self.settings6.active==True:
            ltm = time.asctime(time.localtime(time.time()))
            self.lbl81.text+=ltm
        else:
            self.lbl81.text = "On since: "

def switch_callback1(switchObject, switchValue):
        print('Value of SW1: ', switchValue)
def switch_callback2(switchObject, switchValue):
        print('Value of SW2: ', switchValue)
def switch_callback3(switchObject, switchValue):
        print('Value of LED1: ', switchValue)
def switch_callback4(switchObject, switchValue):
        print('Value of LED2: ', switchValue)

#class myApp(App):
#    def build(self):
 #       return BoxLayout()

if __name__=="__main__":
    myApp=BoxLayoutApp()
    myApp.run()
   # myApp().run()
```