**Project Development Phase**

**Sprint – 4**

|  |  |
| --- | --- |
| Team ID | PNT2022TMID01003 |
| Project Name | Hazardous Area Monitoring for Industrial Plant powered by IoT |
| Maximum Marks |  |

**Task:**

A mobile application for monitoring the Environment parameters around the region of an industry has been developed using MIT App Inventor.

**Screens Information:**

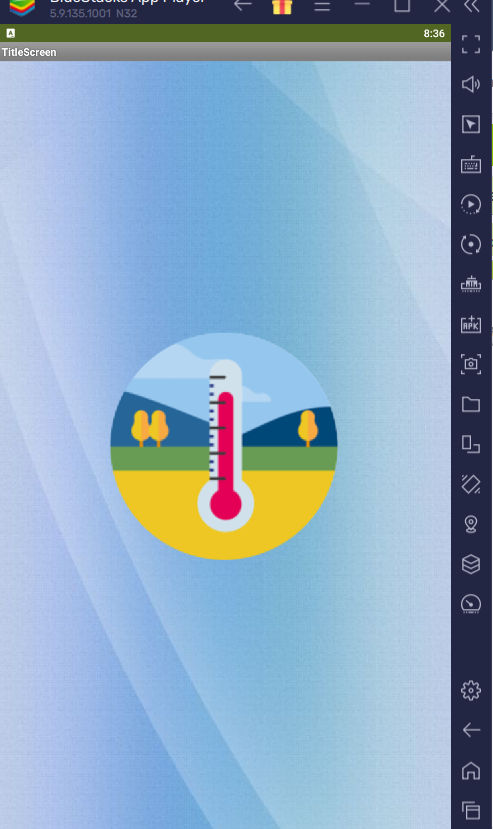
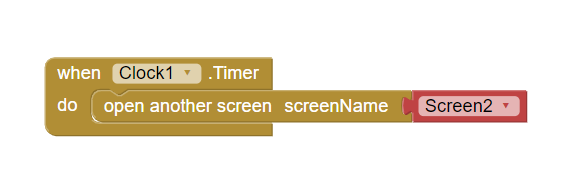
1. **Screen – 1:** It is the entry screen of the mobile application and will be displayed only for 2000 milli-seconds.

2. **Screen – 2:** It is the login page of the application. Each user has their own user id and password, which is known only to them. After validating the credential, the user can access the data captured by the placed device.

3. **Screen – 3:** Environmental parameters in the area of the industry like temperature is obtained via sensors and is sent to the mobile device.

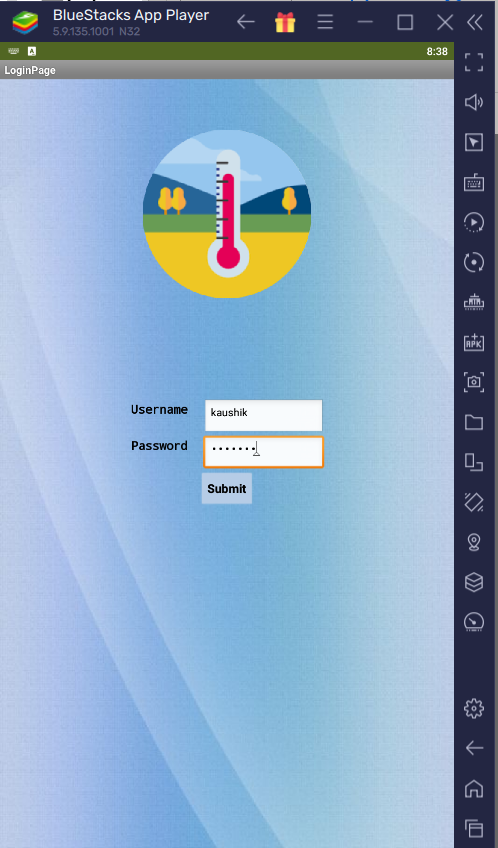
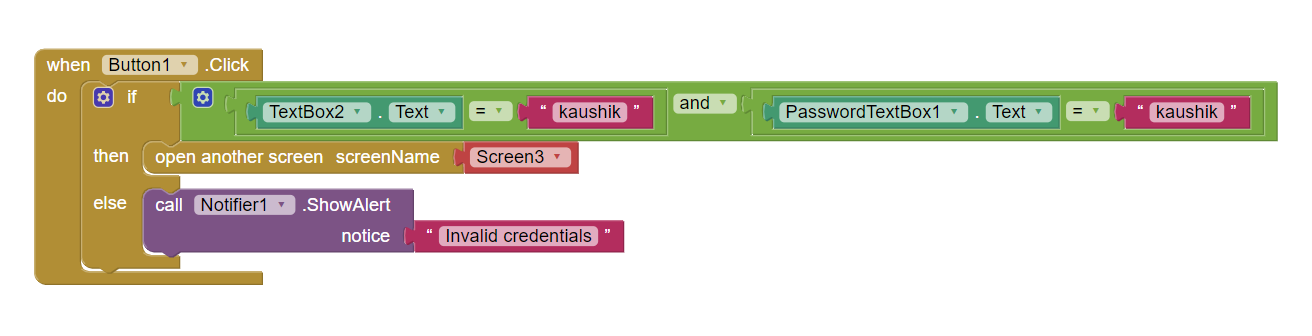
**Screen 1:**

**Designer & Blocks**

**Screen 2:**

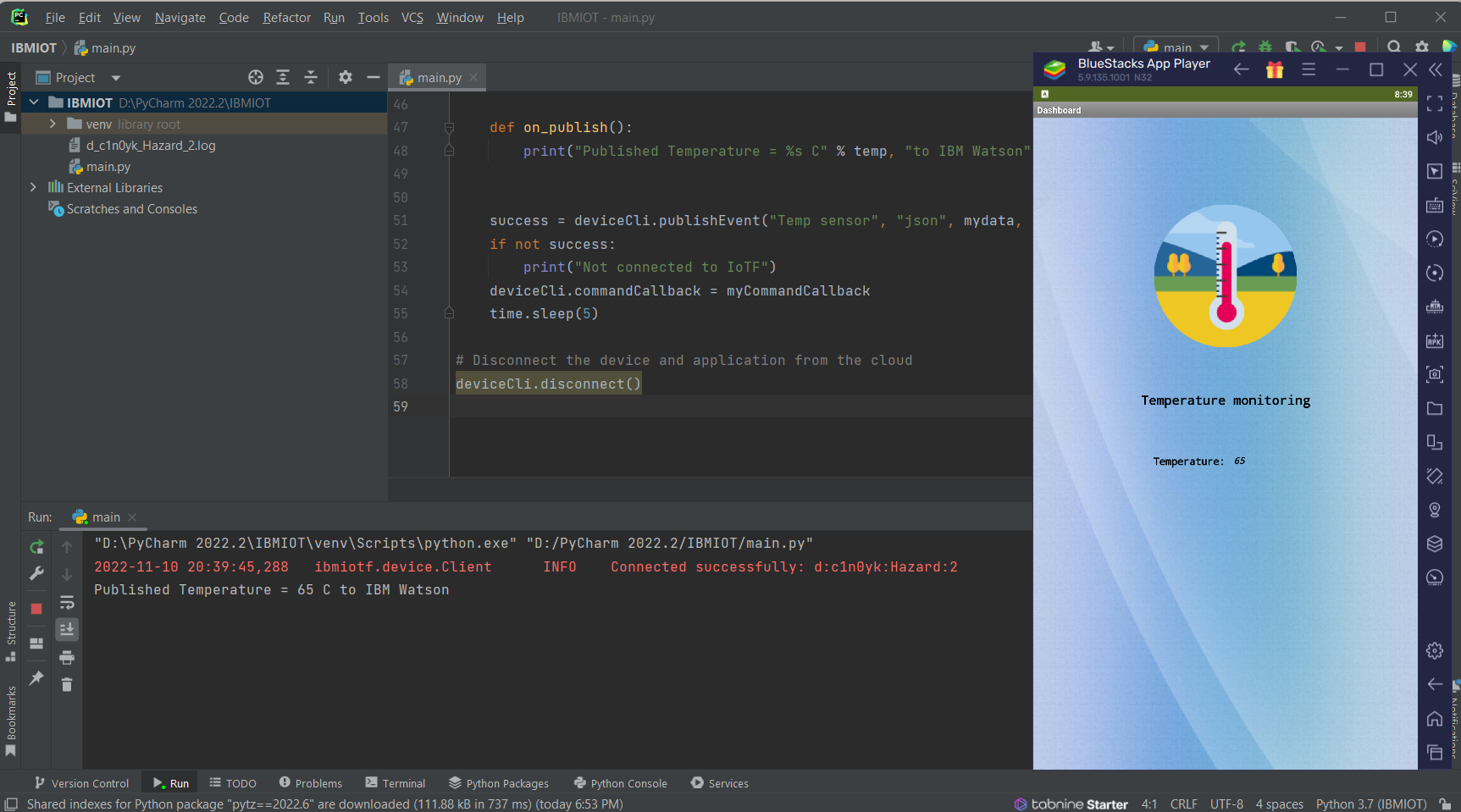
**Designer & Blocks**

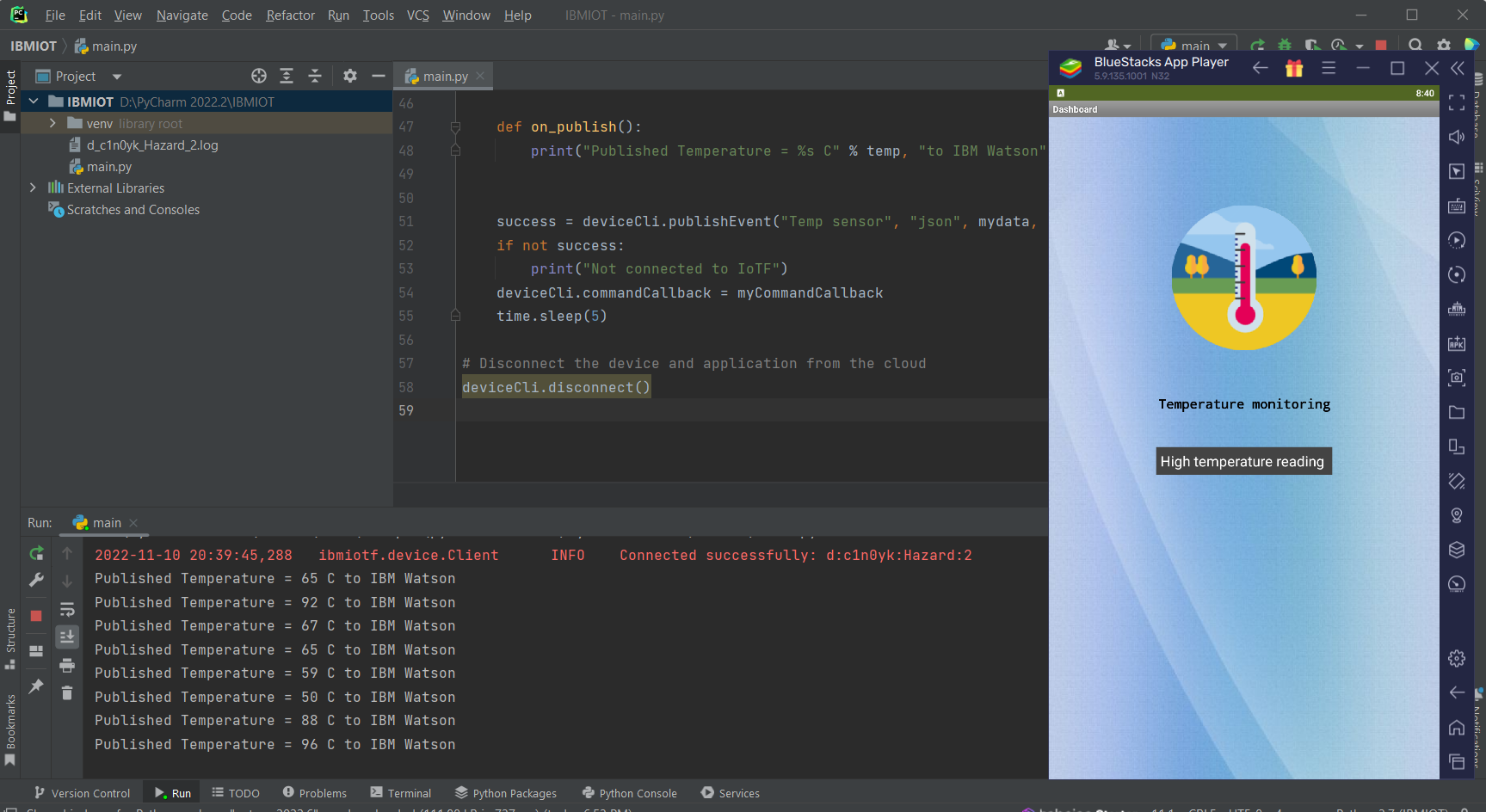
**Screen 3:**

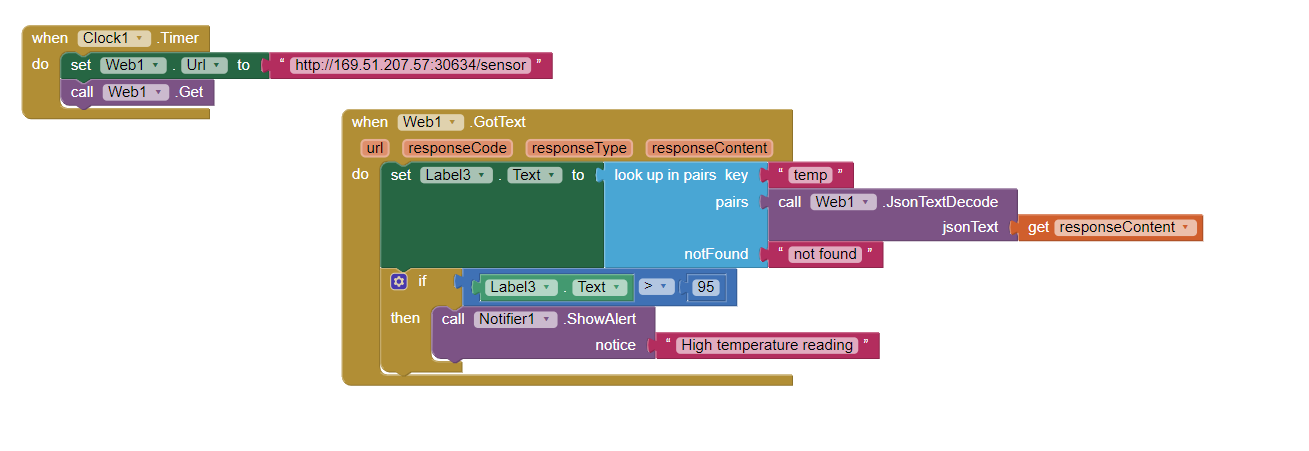
**Designer & Blocks**

*Case 1 (When the temperature is within limit):*



*Case 2 (When temperature exceeds normal (95 C) value):*





**Source code:**

import time  
import sys  
import ibmiotf.application  
import ibmiotf.device  
import random  
  
# Provide your IBM Watson Device Credentials  
organization = "c1n0yk"  
deviceType = "Hazard"  
deviceId = "2"  
authMethod = "token"  
authToken = "123456789"  
  
  
# Initialize GPIO  
def myCommandCallback(cmd):  
 print(cmd)  
 print("Command received: %s" % cmd.data['command'])  
 status = cmd.data['command']  
 if status == "lighton":  
 print("led is on")  
 elif status == "lightoff":  
 print("led is off")  
 else:  
 print("please send proper command")  
  
  
try:  
 deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod,  
 "auth-token": authToken}  
 deviceCli = ibmiotf.device.Client(deviceOptions)  
# ..............................................  
  
except ibmiotf.ConnectionException as e:  
 print("Caught exception connecting device: %s" % str(e))  
 sys.exit()  
deviceCli.connect()  
  
while True:  
 # Get Sensor Data from DHT11  
  
 temp = random.randint(50, 100)  
  
 mydata = {'temp': temp}  
  
  
 def on\_publish():  
 print("Published Temperature = %s C" % temp, "to IBM Watson")  
  
  
 success = deviceCli.publishEvent("Temp sensor", "json", mydata, qos=0, on\_publish=on\_publish)  
 if not success:  
 print("Not connected to IoTF")  
 deviceCli.commandCallback = myCommandCallback  
 time.sleep(5)  
  
# Disconnect the device and application from the cloud  
deviceCli.disconnect()