

PROJECT DEVELOPMENT PHASE SPRINT-1

Team ID	PNT2022TMID41751
Project Name	IoT- Based Smart Crop Protection System For Agriculture

PYTHON CODE AND LOGIN FORM

Python Code:

```
import random
import
ibmiotf.application
import ibmiotf.device
from time import sleep
import sys

#IBM Watson Device Credentials.
organization = "Jy712s" deviceType = "leena"
deviceId = "leena123" authMethod = "token"
authToken = "123456789" def
myCommandCallback(cmd): print("Command
received: %s" % cmd.data['command'])
status=cmd.data['command'] if
status=="sprinkler_on": print ("sprinkler is ON")
else :
print ("sprinkler is OFF")
#print(cmd)
try:
```

```
deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod,
"auth-token": authToken} deviceCli = ibmiotf.device.Client(deviceOptions) except Exception as e:
```

```
    print("Caught exception connecting device: %s" % str(e))
```

```
    sys.exit()
```

```
#Connecting to IBM watson.
```

```
deviceCli.connect()
```

```
while True:
```

```
#Getting values from sensors.
```

```
    temp_sensor = round( random.uniform(0,80),2) PH_sensor = round(random.uniform(1,14),3) camera
= ["Detected","Not Detected","Not Detected","Not Detected","Not Detected","Not Detected",]
camera_reading = random.choice(camera) flame = ["Detected","Not Detected","Not Detected","Not
Detected","Not Detected","Not Detected",] flame_reading = random.choice(flame) moist_level =
round(random.uniform(0,100),2) water_level = round(random.uniform(0,30),2)
```

```
#storing the sensor data to send in json format to cloud.
```

```
    temp_data = { 'Temperature' : temp_sensor }
```

```
PH_data = { 'PH Level' : PH_sensor } camera_data
```

```
= { 'Animal attack' : camera_reading} flame_data
```

```
= { 'Flame' : flame_reading } moist_data = {
```

```
'Moisture Level' : moist_level} water_data = {
```

```
'Water Level' : water_level}
```

```

# publishing Sensor data to IBM Watson for every 5-10 seconds.    success =
deviceCli.publishEvent("Temperature sensor", "json", temp_data, qos=0)

    sleep(1)
if success
print ("
.....
.....publish ok.....
..... ")

    print ("Published Temperature = %s C" % temp_sensor, "to IBM Watson") success
= deviceCli.publishEvent("PH sensor", "json", PH_data, qos=0)

    sleep(1) if
success:

    print ("Published PH Level = %s" % PH_sensor, "to IBM Watson")

success = deviceCli.publishEvent("camera", "json", camera_data, qos=0)
sleep(1) if success:

    print ("Published Animal attack %s " % camera_reading, "to IBM Watson")    success =
deviceCli.publishEvent("Flame sensor", "json", flame_data, qos=0)

    sleep(1)
if success:

print ("Published Flame %s " % flame_reading, "to IBM Watson")

    success = deviceCli.publishEvent("Moisture sensor", "json", moist_data, qos=0)

    sleep(1)
if success:

```

```

print ("Published Moisture Level = %s " % moist_level, "to IBM Watson")

success = deviceCli.publishEvent("Water sensor", "json", water_data, qos=0)

    sleep(1)

if success:

print ("Published Water Level = %s cm" % water_level, "to IBM Watson") print
("")

```

#Automation to control sprinklers by present temperature and to send alert message to IBM Watson.

```

if (temp_sensor > 35):

    print("sprinkler-1 is ON")

    success = deviceCli.publishEvent("Alert1", "json", { 'alert1' : "Temperature(%s) is high, sprinklers
are turned ON"

%temp_sensor } ,

qos=0)

    sleep(1) if

success: print(

'Published

alert1 : ',

"Temperatur

e(%s) is high,

sprinklers are

turned

```

```
ON" %temp_senso
```

```
r,"to IBM
```

```
Watson")
```

```
    print("")
```

```
else:    print("sprinkler-1
```

```
is OFF") print("")
```

```
#To send alert message if farmer uses the unsafe fertilizer to crops.
```

```
if (PH_sensor > 7.5 or PH_sensor < 5.5):
```

```
success = deviceCli.publishEvent("Alert2", "json",{ 'alert2' : "Fertilizer PH level(%s) is not safe,use other  
fertilizer" %PH_sensor }, qos=0) sleep(1)    if success:
```

```
    print('Published alert2 : ' , "Fertilizer PH level(%s) is not safe,use other fertilizer" %PH_sensor,"to IBM  
Watson")
```

```
    print("")
```

```
#To send alert message to farmer that animal attack on crops.
```

```
if (camera_reading == "Detected"):
```

```
    success = deviceCli.publishEvent("Alert3", "json", { 'alert3' : "Animal attack on crops detected" },  
qos=0)
```

```
    sleep(1)
```

```
if success: print('Published alert3 : ' , "Animal attack on crops detected","to IBM Watson","to IBM  
Watson") print("")
```

```
#To send alert message if flame detected on crop land and turn ON the splinkers to take immediate  
action.
```

```
if (flame_reading == "Detected"):
```

```
print("sprinkler-2 is ON") success = deviceCli.publishEvent("Alert4", "json", { 'alert4' : "Flame is detected crops are in danger,sprinklers turned ON" }, qos=0)
```

```
sleep(1) if
```

```
success:
```

```
print( 'Published alert4 : ' , "Flame is detected crops are in danger,sprinklers turned ON","to IBM Watson")
```

```
print("")
```

```
else:
```

```
print("sprinkler-2 is OFF")
```

```
print("")
```

```
#To send alert message if Moisture level is LOW and to Turn ON Motor-1 for irrigation.
```

```
if (moist_level < 20):
```

```
print("Motor-1 is ON") success = deviceCli.publishEvent("Alert5", "json", { 'alert5' : "Moisture level(%s) is low,
```

```
Irrigation started" %moist_level }, qos=0)
```

```
sleep(1) if success: print('Published alert5 : ' , "Moisture level(%s) is low, Irrigation started"
```

```
%moist_level,"to IBM Watson" ) print("") else: print("Motor-1 is OFF") print("")
```

```
#To send alert message if Water level is HIGH and to Turn ON Motor-2 to take water out.
```

```
if (water_level > 20): print("Motor-2 is ON")
```

```
success = deviceCli.publishEvent("Alert6", "json", { 'alert6' : "Water level(%s) is high, so motor is ON to take water out " %water_level }, qos=0)
```

```
sleep(1)
```

```
if success:
```

```
print('Published alert6 : ' , "water level(%s) is high, so motor is ON to take water out " %water_level,"to  
IBM Watson" )
```

```
print("")
```

```
else:
```

```
print("Motor-2 of OFF") print("")
```

```
    #command recived by farmer
```

```
deviceCli.commandCallback = myCommandCallback #
```

```
Disconnect the device and application from the cloud
```

```
deviceCli.disconnect()
```

Login-form:

```

<!DOCTYPE html>
<!-- Created By CodingNepal -->
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Login Form</title>
    <link rel="stylesheet" href="login-style.css">
    <meta name="viewport" content="width=device-width,
initialscale=1.0">    </head>
  <body>
    <div class="wrapper">
      <div class="title-text">
        <div class="title login">
          Login Form
        </div>
      </div>
      <div class="form-container">
        <div class="form-inner">
          <form action="#" class="login">
            <div class="field">
              <input type="text" placeholder="Email Address"
required>
            </div>
            <div class="field">
              <input type="password" placeholder="Password"
required>
            </div>
            <div class="pass-link">
              <a href="#">Forgot password?</a>
            </div>
            <div class="field btn">
              <div class="btn-layer"></div>
              <input type="submit" value="Login">
            </div>
            <div class="signup-link">

```




```
Not a member? <a href="Register.html">Signup
now</a>
    </div>
  </form>
</div>
</div>
</div>
</body>
</html>
```

Registration form:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,
initialscale=1.0">
  <title>Registration form</title>
  <link rel="stylesheet" href="login-style.css">
</head>

<body>
  <div class="wrapper">
    <div class="title signup">
      Signup Form
    </div>
    <div class="form-container">
      <div class="form-inner">
        <form action="#" class="signup">
          <div class="field">
            <input type="text" placeholder="Email Address"
required>
          </div>
          <div class="field">
            <input type="password" placeholder="Password">
          </div>
        </form>
      </div>
    </div>
  </div>
</body>
</html>
```

```

required>
        </div>
        <div class="field">
            <input type="password" placeholder="Confirm
password" required>
        </div>
        <div class="field btn">
            <div class="btn-layer"></div>
            <input type="submit" value="Signup">
        </div>
        <div class="signup-link">
already have an account <a href="index.html">Login
now</a>
        </div>
    </form>
</div>
</div>
</div>
</body>
</html>

```

Css-file:

```

@import
url('https://fonts.googleapis.com/css2?family=Source+Sans+Pro:ital,wgh

```

```
t@0,200;0,300;0,400;1,400&display=swap');
*{ margin: 0; padding: 0; box-sizing:
border-box; font-family: 'Source Sans Pro',
sans-serif;;
} html,body{
display: grid;
height: 100%;
```

```
width: 100%;
place-items: center; background: -webkit-linear-
gradient(left, #f8b7cd, #c8e7f5);
```

[REDACTED]

[REDACTED]

```

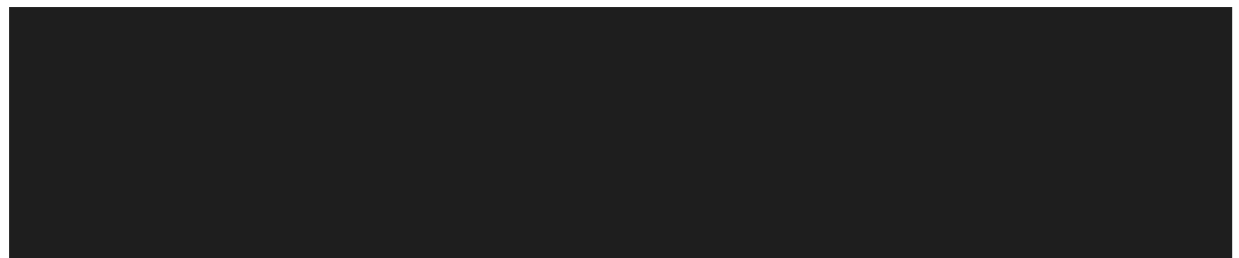
::selection{
  /* background: #fa4299; */
color: #fff;
}
.wrapper{  overflow: hidden;  background:
#fff;  width: 30%;  padding: 30px;
border-radius: 5px;  box-shadow: 0px 15px
20px rgba(0,0,0,0.1); }
.wrapper .title-text{
display: flex;  width:
200%;
}
.wrapper .title{  width: 50%;  font-size: 35px;  font-
weight: 600;  text-align: center;  transition: all 0.6s
cubic-bezier(0.68,-0.55,0.265,1.55); }
.wrapper .slide-controls{
position: relative;
display: flex;  height:
50px;  width: 100%;
overflow: hidden;

  margin: 30px 0 10px 0;  justify-
content: space-between;  border: 1px
solid lightgrey;  border-radius: 5px;
}
.slide-controls .slide{


```



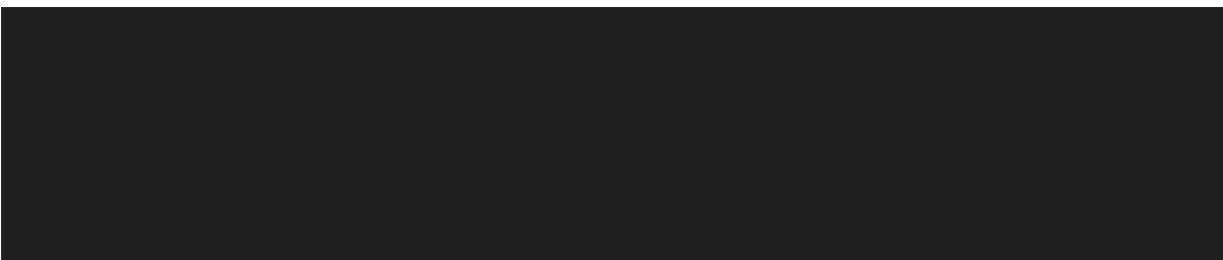
```
height: 100%;  
width: 100%;  color:  
#fff;  font-size:  
18px;
```



```
font-weight: 500; text-align: center; line-height: 48px; cursor: pointer; z-index: 1; transition: all 0.6s ease;
} input[type="radio"]{
display: none;
}
#signup:checked ~ .slider-tab{
left: 50%;
}
#signup:checked ~ label.signup{
color: #fff; cursor: default;
user-select: none;
}
#signup:checked ~ label.login{
color: #000;
}
#login:checked ~ label.signup{
color: #000;
}
#login:checked ~ label.login{
cursor: default; user-select: none;
}
.wrapper .form-container{
```

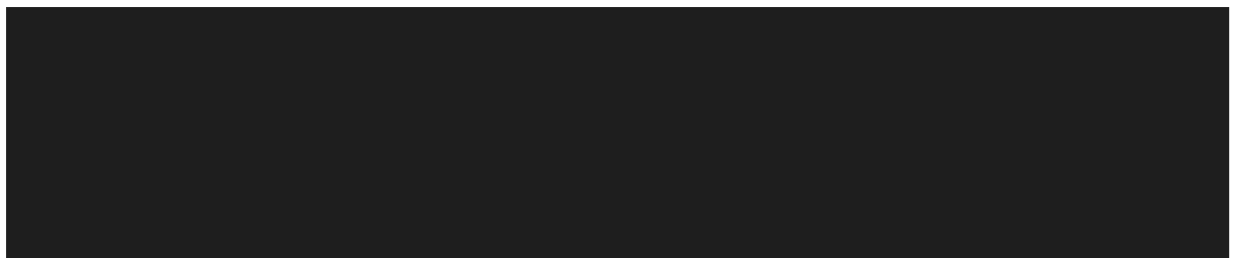
```
    width: 100%;  
overflow: hidden; }  
.form-container .form-inner{  
display: flex; width: 200%;  
}  
.form-container .form-inner form{ width: 50%;  
transition: all 0.6s cubic-bezier(0.68,-0.55,0.265,1.55);
```



```
}
.form-inner form .field{
height: 50px;  width:
100%;  margin-top:
20px;
}
.form-inner form .field input{
height: 100%;  width: 100%;
outline: none;  padding-left:
15px;  border-radius: 5px;
border: 1px solid lightgrey;
border-bottom-width: 2px;
font-size: 17px;
  transition: all 0.3s ease;
}
.form-inner form .field input:focus{  border-
color: #fc83bb;
  /* box-shadow: inset 0 0 3px #fb6aae; */
}
.form-inner form .field input::placeholder{
color: #999;  transition: all 0.3s ease;
} form .field
input:focus::placeholder{  color:
#b3b3b3;
}
.form-inner form .pass-link{
```



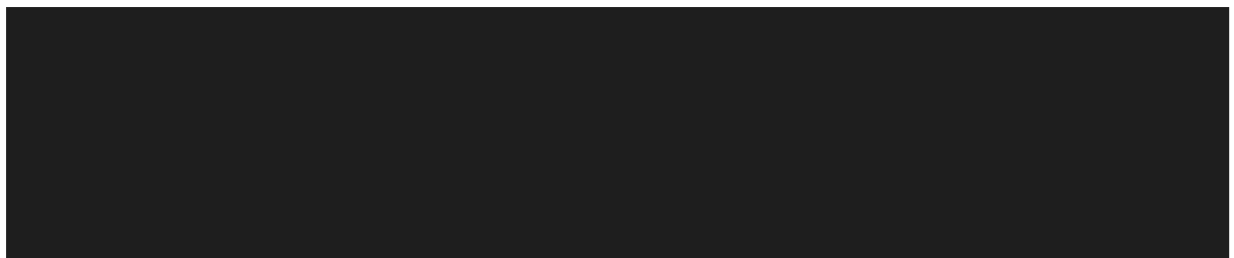
```
    margin-top: 5px;
}
.form-inner form .signup-link{
text-align: center; margin-top:
30px;
}
.form-inner form .pass-link a,
.form-inner form .signup-link a{
color: #fa4299; text-
decoration: none;
```



```
}
.form-inner form .pass-link a:hover, .form-inner
form .signup-link a:hover{  text-decoration:
underline;
} form .btn{
height: 50px;
width: 100%;
border-radius: 5px;
position: relative;
overflow: hidden;
    } form .btn .btn-layer{  height: 100%;  width: 300%;
position: absolute;  left: -100%;  background: -webkit-
linear-gradient(left,#71d9e3,#fc95f4);  border-radius: 5px;
    transition: all 0.4s ease;;
} form .btn:hover .btn-
layer{  left: 0;
} form .btn
input[type="submit"]{  height:
100%;  width: 100%;  z-index:
1;
```



```
    position: relative;  
background: none;  
border: none;  
color: #fff;  
padding-left: 0;  
border-radius: 5px;
```



```
font-size: 20px; font-  
weight: 500; cursor:  
pointer;  
}
```

Js-file:

```
const loginText = document.querySelector(".title-text .login");  
const loginForm = document.querySelector("form.login"); const  
loginBtn = document.querySelector("label.login"); const signupBtn  
= document.querySelector("label.signup"); const signupLink =  
document.querySelector("form .signup-link a"); signupBtn.onclick  
= (() => { loginForm.style.marginLeft = "-50%";  
loginText.style.marginLeft = "-50%";  
}); loginBtn.onclick = (() => {  
loginForm.style.marginLeft = "0%";  
loginText.style.marginLeft = "0%";  
}); signupLink.onclick = (()  
=> { signupBtn.click();  
return false;  
});
```