Smart Farmer - IoT Enabled Smart Farming Application

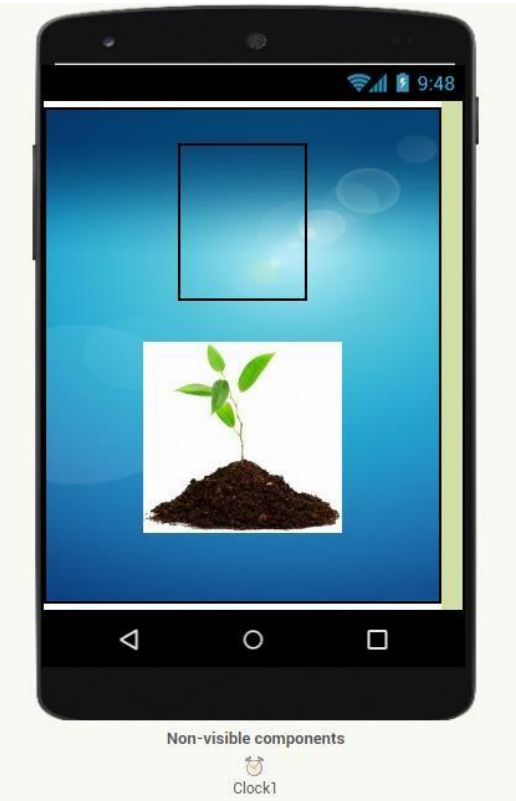
SPRINT-4

Team ID	PNT2022TMID35924
Project name	Smart farmer- IoT enabled smart farming

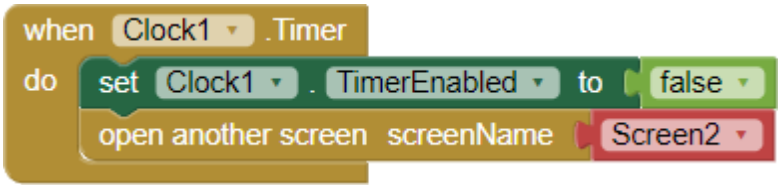
In this session, we have developed the App for our project using MIT App Inventor, which displays the following field parameters – Moisture, Temperature, Humidity, Flame, Nitrogen, Phosphorous, Potassium. It also displays the Motor ON and OFF switches to control the water sprinklers located in the field from our Mobile itself. It consists of three screens whose screenshots are attached below:

SCREEN 1:

Front End:



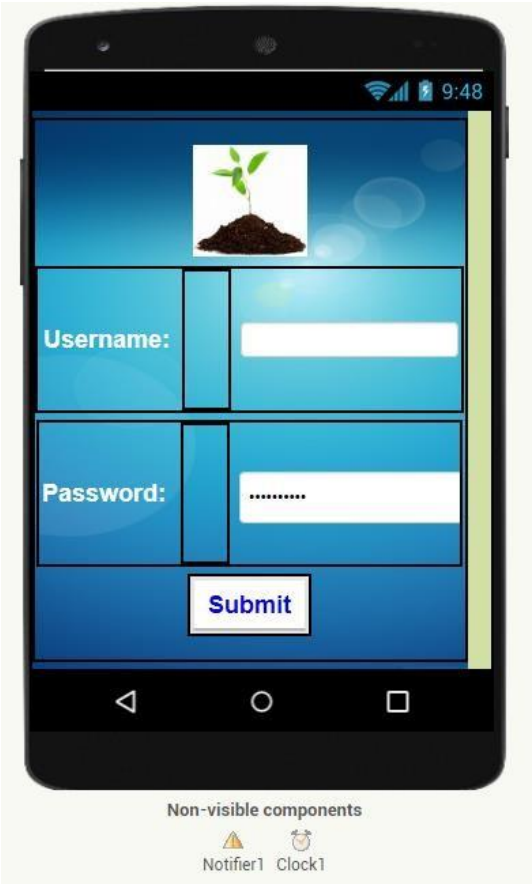
Back End:



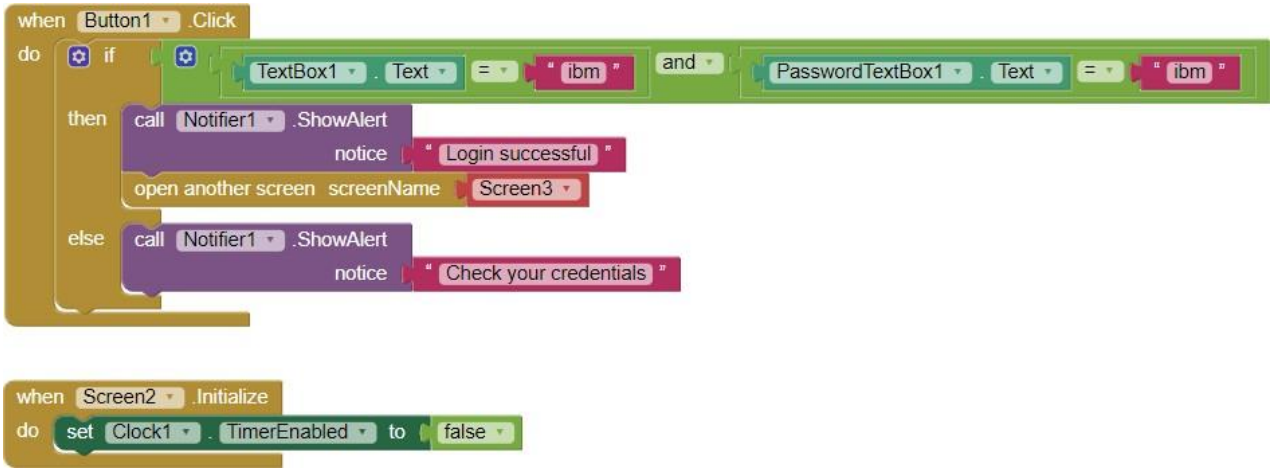
The above screenshot shows our screen 1. When the App is opened, this screen will be displayed for 3s then it moves to our login page. We have used Vertical Arrangement, Horizontal Arrangement and Clock components.

SCREEN 2:

Front End:



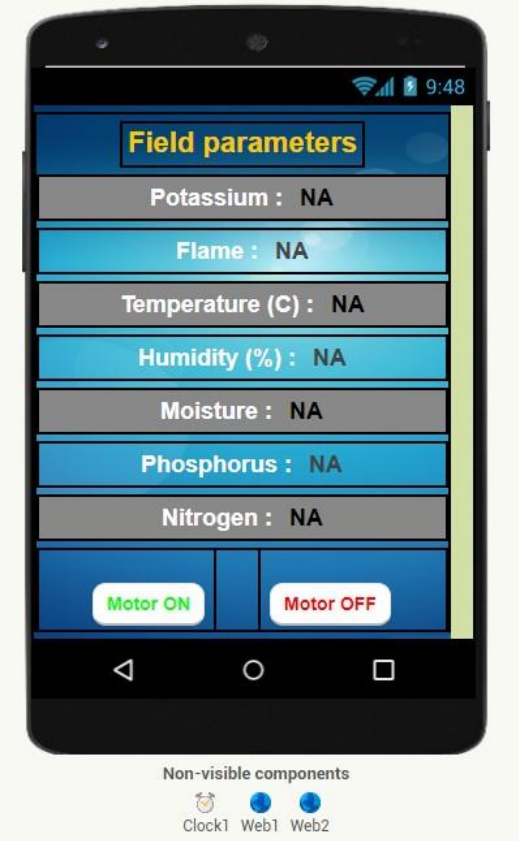
Back End:



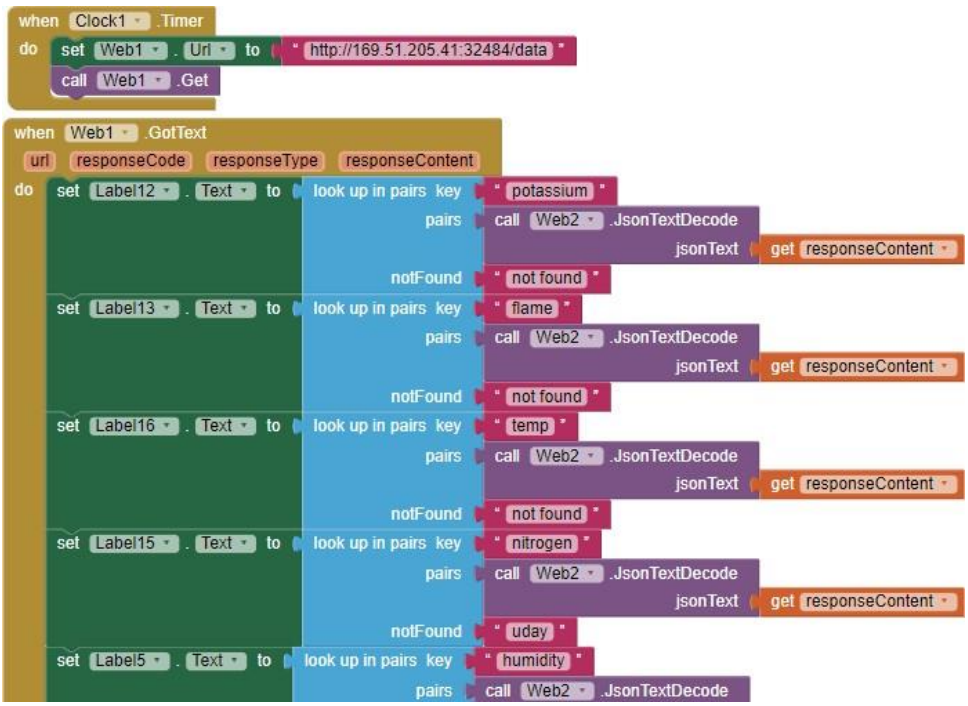
This shows our screen 2. In this, we have designed our login page. When the user feeds the correct credentials, they will be moved to the next page. And if the login credentials are wrong, they will be stuck in the same page until they enter the correct details. We have used vertical and horizontal arrangements, labels, textbox, password textbox, button, notifier and clock components in this.

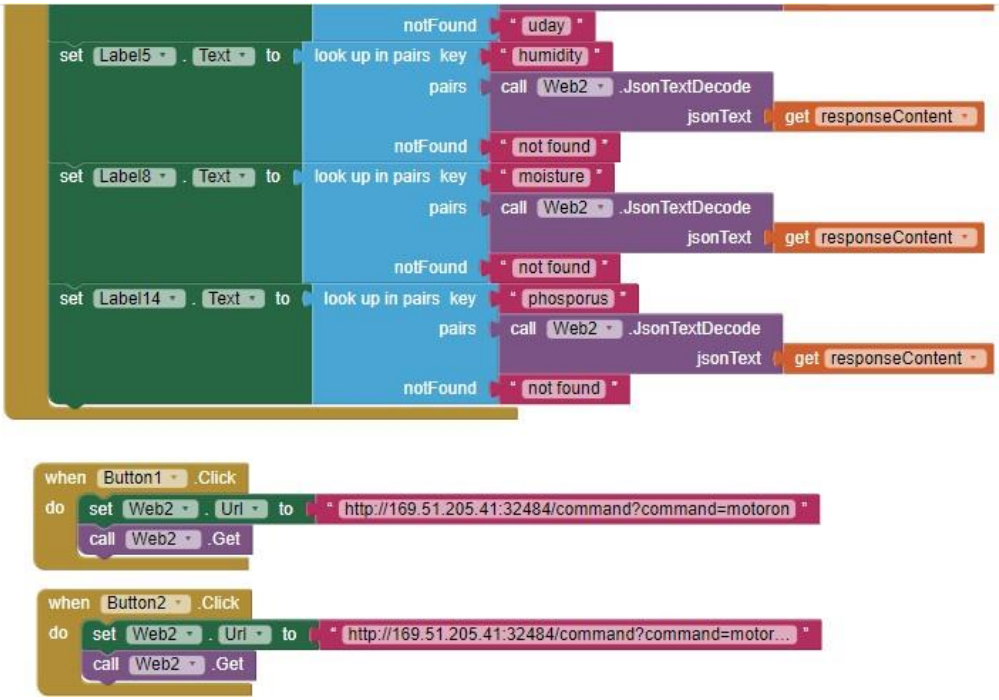
SCREEN 3:

Front End:



Back End:



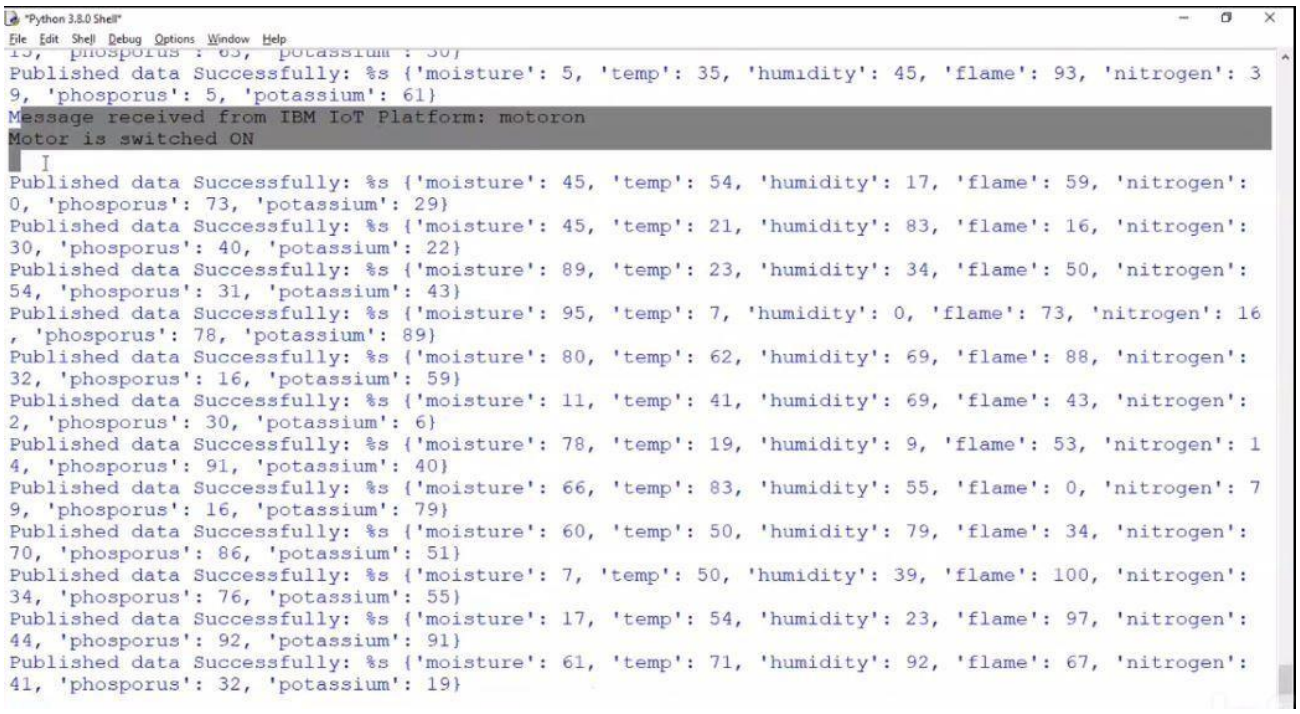


This Screen is our final one. This displays all our field parameters, the inputs of which are delivered by the Node-RED Software. When the moisture reading drops below a certain level or the flame threshold is reached, an 'alert' SMS is sent to the user. The Motor ON/OFF switches are also used to control the field irrigation system. We have used vertical and horizontal arrangement, label, button, clock and web components.

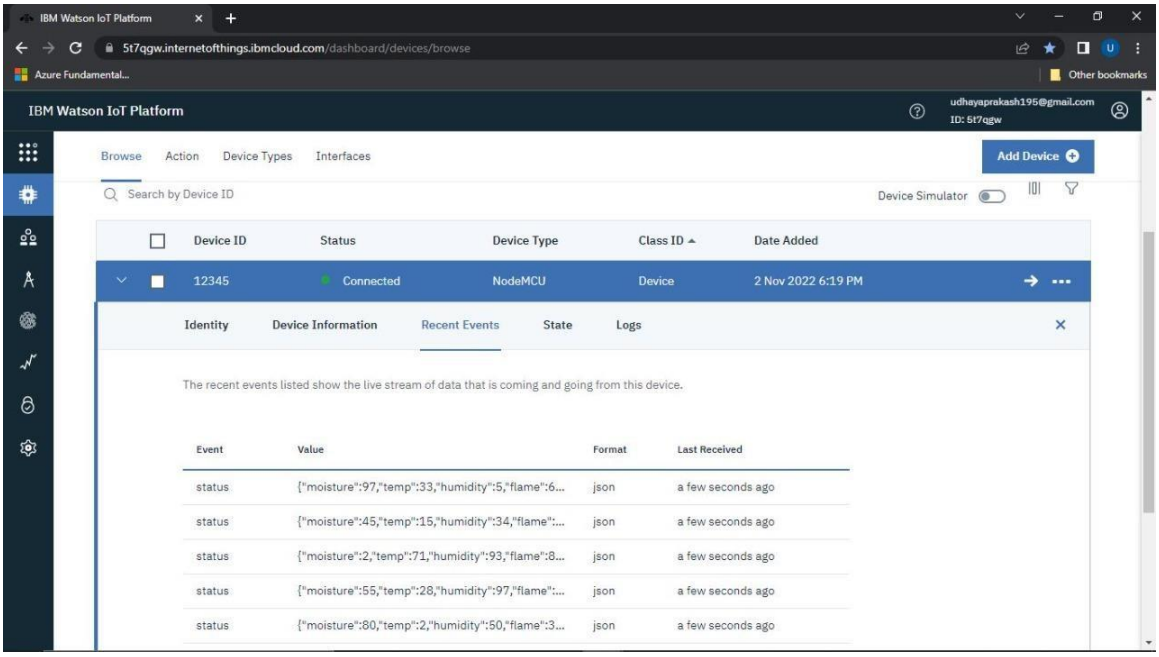
OUTPUT:

The following output flow is observed:

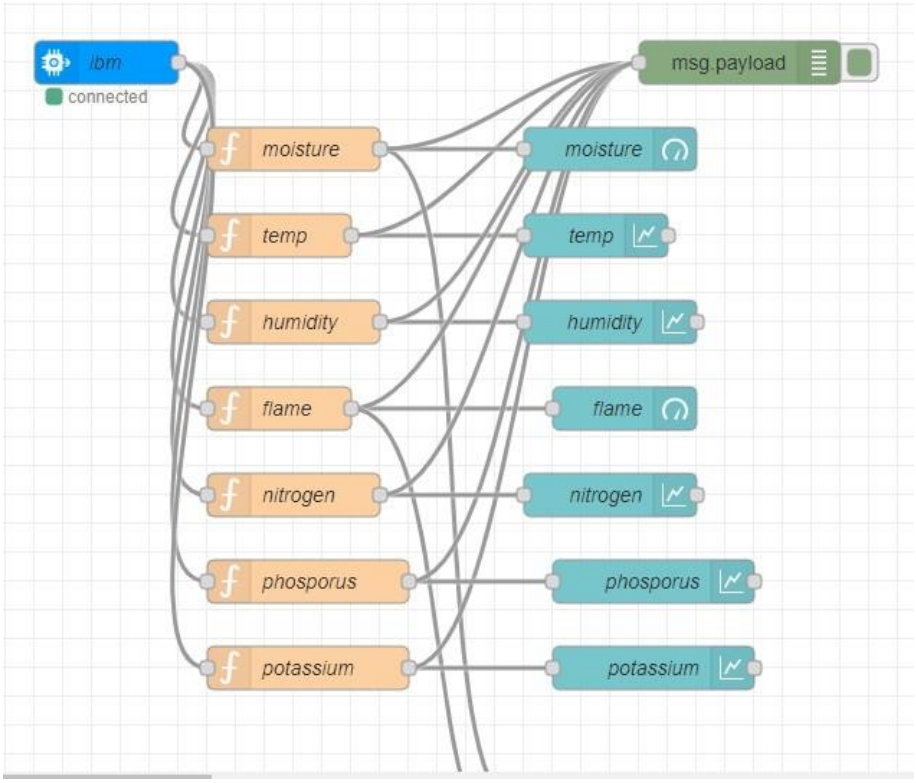
Python IDLE:

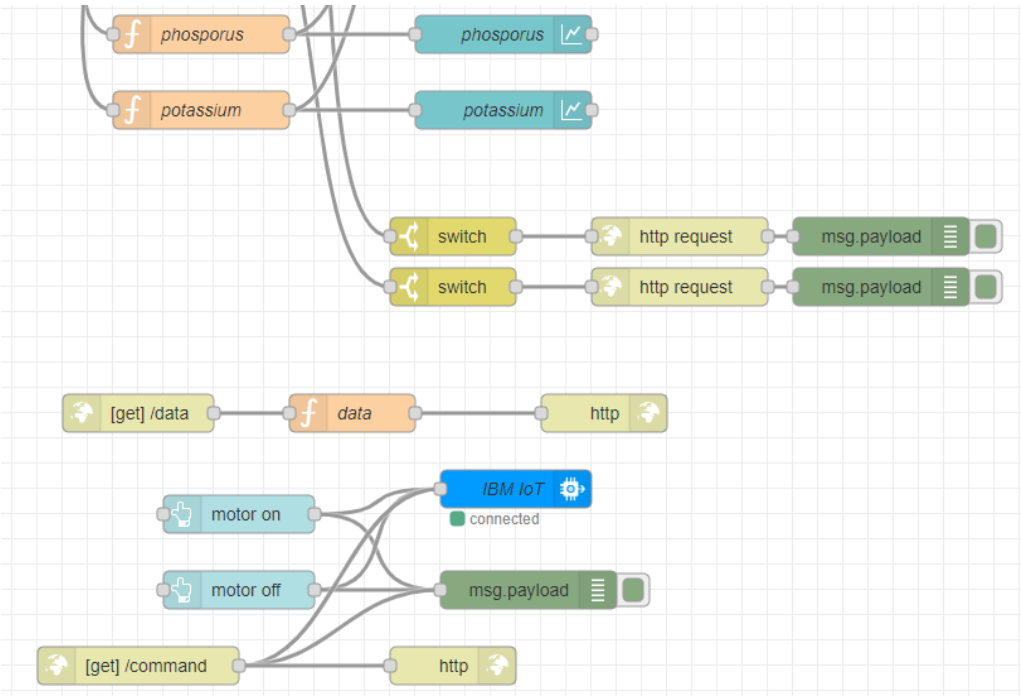


IBM Cloud:

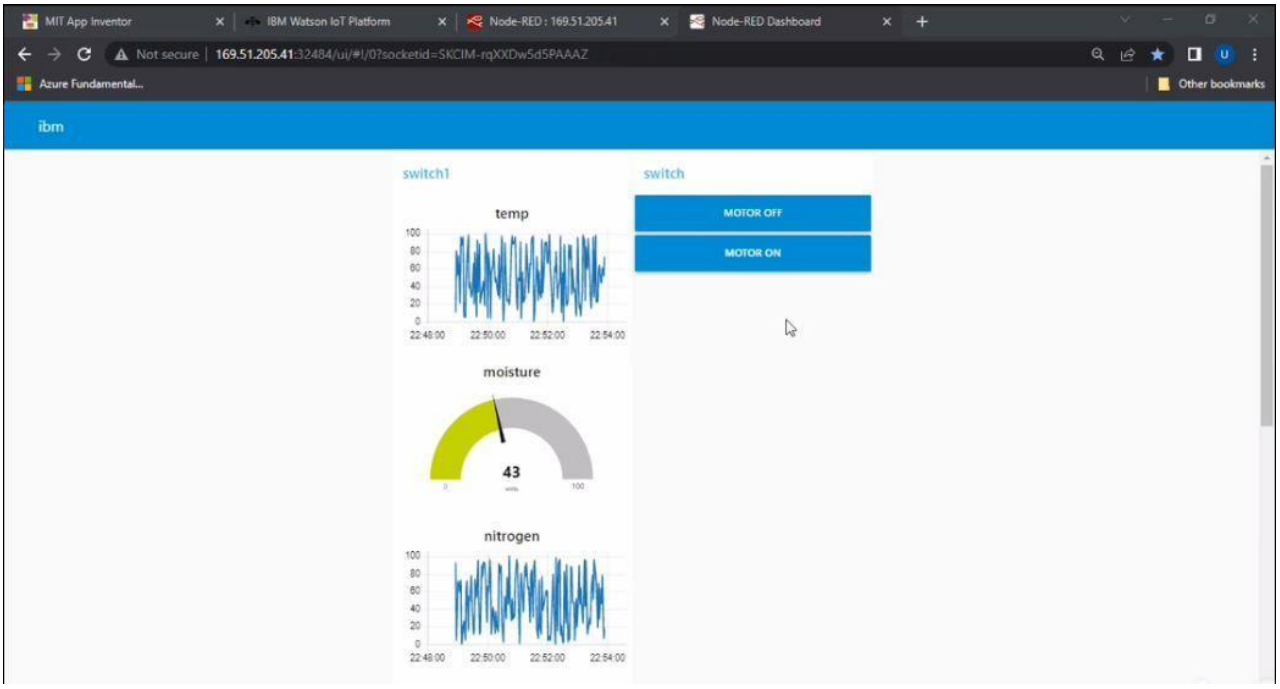


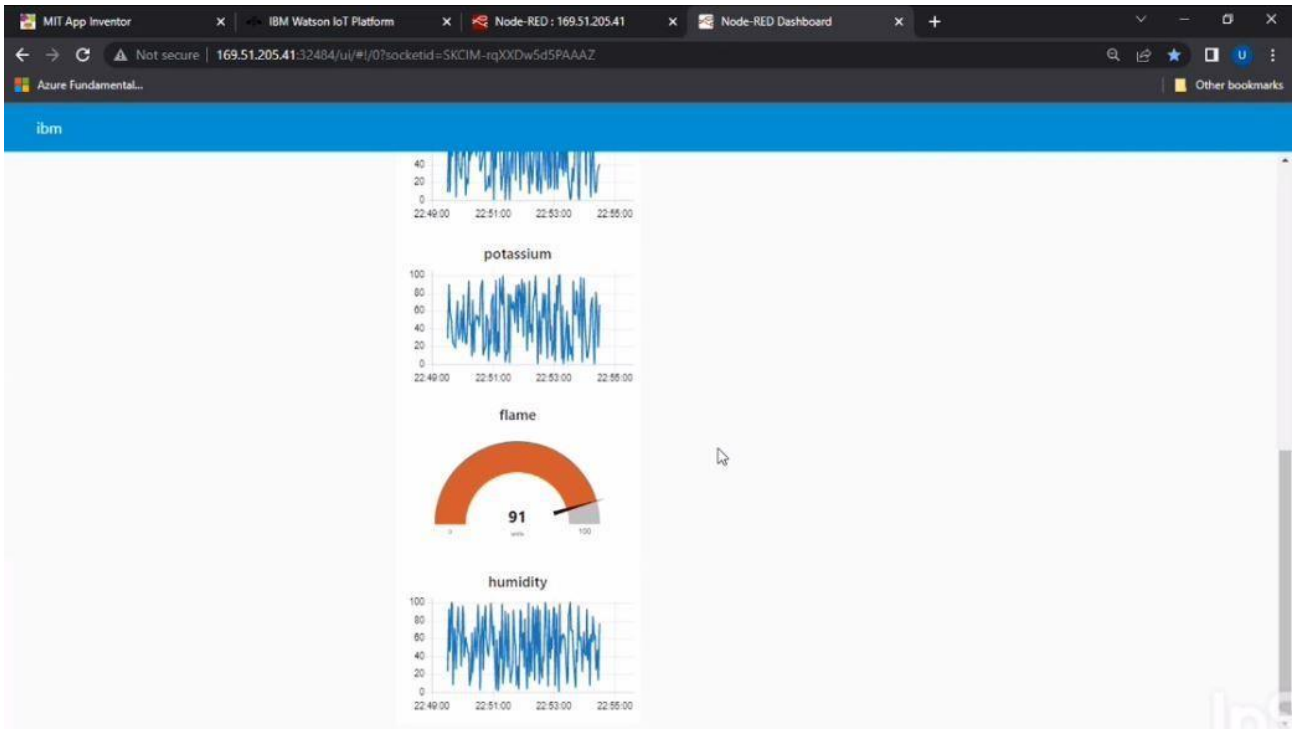
Node-RED Flow:




phosporus
phosporus
potassium
potassium
switch
http request
msg.payload
switch
http request
msg.payload
[get] /data
data
http
IBM IoT
connected
motor on
motor off
msg.payload
[get] /command
http

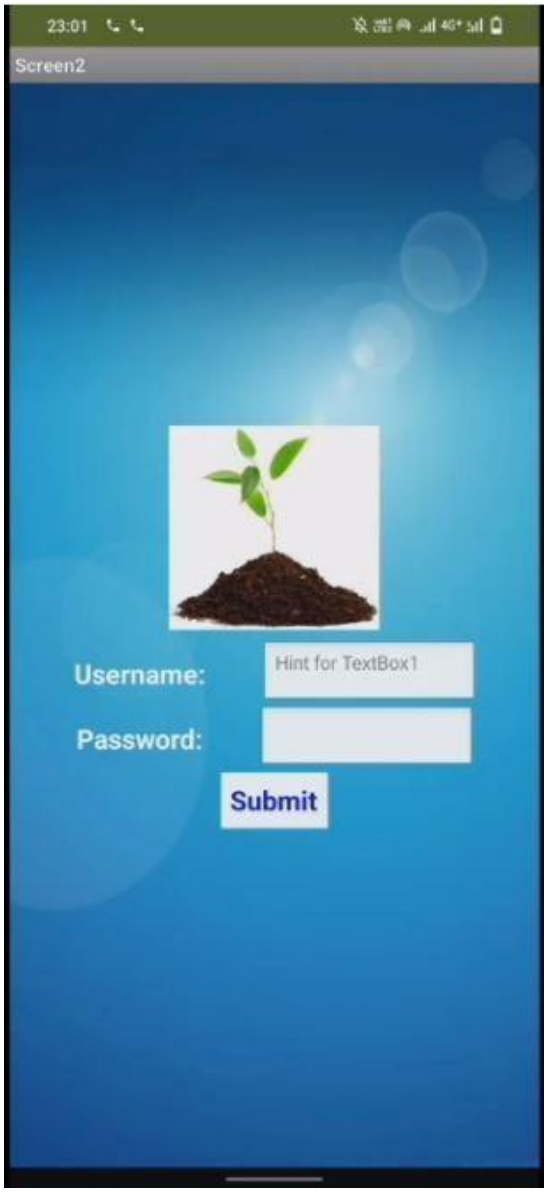
Web-UI:


MIT App Inventor
IBM Watson IoT Platform
Node-RED : 169.51.205.41
Node-RED Dashboard
Not secure
169.51.205.41:32484/ui/#!/0?socketid=SKCIM-rqXXDw5d5PAAAZ
Azure Fundamental...
Other bookmarks
ibm
switch1
temp
switch
MOTOR OFF
MOTOR ON
moisture
43
nitrogen

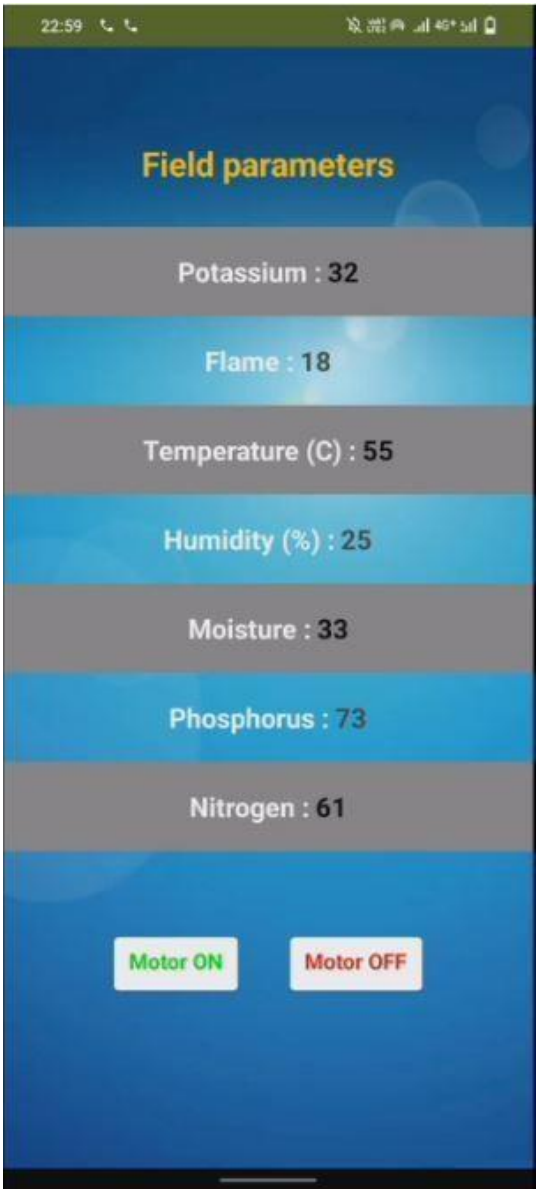


App Display:
Screen1:

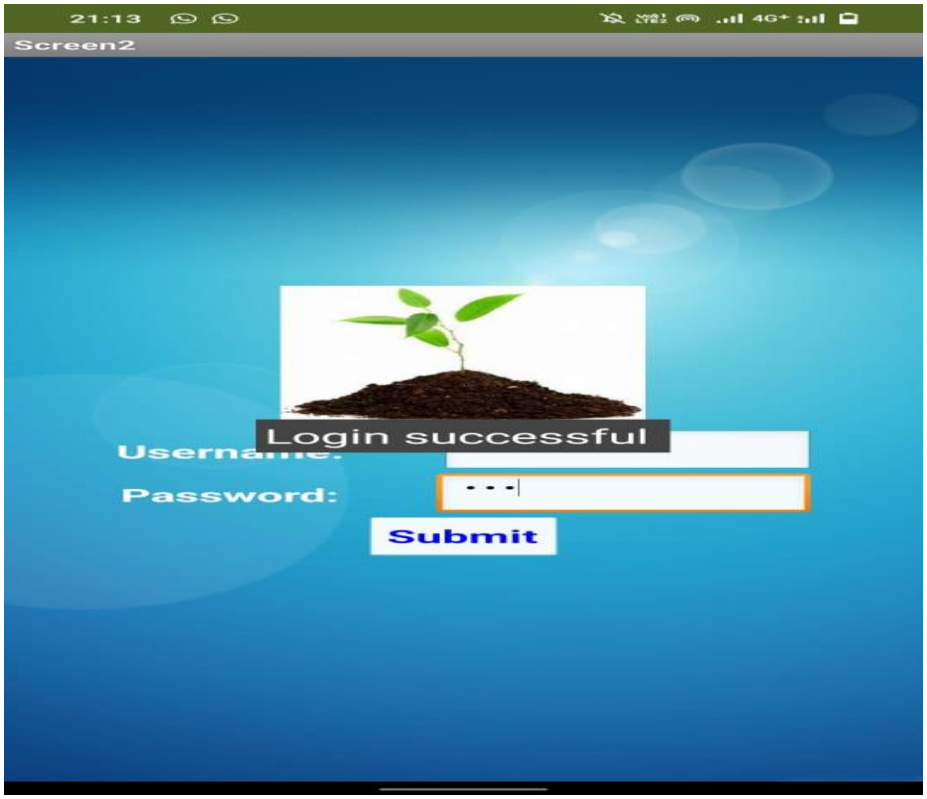
Screen 2:



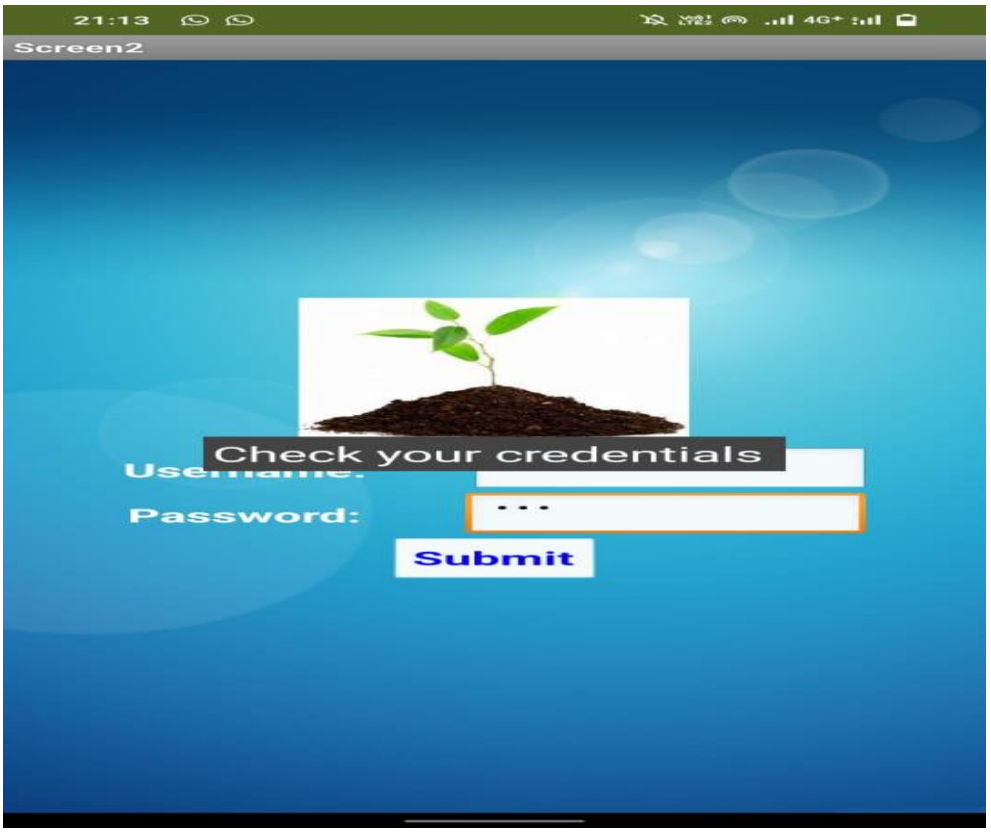
Screen 3:



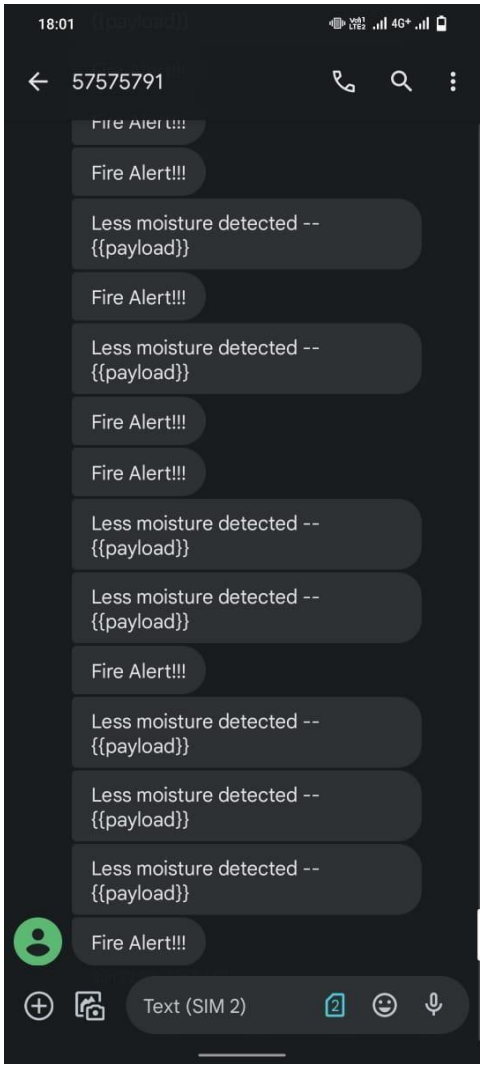
When correct credentials are entered in the app:



When wrong credentials are entered in the app:



SMS ‘Alert’:



Motor ON:

```
Message received from IBM IoT Platform: motoron
Motor is switched ON
```

Motor OFF:

```
Message received from IBM IoT Platform: motoroff
Motor is switched OFF
```