| Test case ID | Feature Type | Component | Test Scenario |
|---|---|---|---|
| TC_OO1 | Functional | IBM cloud | Create the IBM Cloud services which are being used in this project. |
| TC_OO2 | Functional | IBM Cloud | Configure the IBM Cloud services which are being used in completing this project. |
| TC_OO3 | Functional | IBM Watson IoT Platform | IBM Watson IoT platform acts as the mediator to connect the web application to IoT devices, so create the IBM Watson IoT platform. |
| TC_OO4 | Functional | IBM Watson | In order to connect the IoT device to the IBM cloud, create a device in the IBM Watson IoT platform and get the device credentials. |
| TC_OO5 | Functional | IBM Cloud(Node Red) | Configure the connection security and create API keys that are used in the Node-RED service for accessing the IBM IoT Platform. |

| TC_OO6 | Functional | Node Red | Create a Node-RED service. |
|--------|-----------|----------|---------------------------|
| TC_OO7 | Functional | Python 3.7.0 | Develop a python script to publish random sensor data such as temperature, humidity level, soil moisture to the IBM IoT platform |
| TC_OO8 | Functional | Python 3.7.0 | After developing python code, commands are received just print the statements which represent the control of the devices. |
| TC_OO9 | Functional | IBM Cloudant D | Publish Data to The IBM Cloud |
| TC_O10 | Web UI | Node Red & MIT Inventor | Create Web UI in Node- Red |
| TC_O11 | Functional | IBM Cloudant DB | Configure the Node-RED flow to receive data from the IBM IoT platform and also use Cloudant DB nodes to store the received sensor data in the cloudant DB |
| | | | |

| | | | |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

| Date | 17-Nov-22 | |
|---|---|---|
| Team ID | PNT2022TMID24852 | |
| Project Name | IoT Based Smart Crop Protection System | |
| Maximum Marks | 4 marks | |
| **Pre-Requisite** | **Steps To Execute** | **Test Data** |
| IBM Cloud Login ID & Password | 1.Go to IBM Cloud signup page<br>2.Enter e-mail id and other credentials<br>3.Enter a password | https://cloud.ibm.com/login |
| IBM Cloud Login ID & Password | 1.Go to Cloud login<br>2.Enter user ID & Password<br>3.Verify login by the popup display | https://cloud.ibm.com/login |
| IBM Watson IoT Platform Login ID & Password | 1.Login to IBM Cloud<br>2.Click Catalog<br>3.Search IoT and click create<br>4.Go to resource list and search Internet of Things platform<br>5.Press Launch and click Sign in IBM Watson Platform | https://eynrcc.internetofthings.ibmcloud.com/dashboard/devices/browse/add |
| IBM Watson IoT Platform Login ID & Password | 1.Login to IBM Watson Platform<br>2. Click Add Device<br>3.Enter the details and click Finish. Create Device ID & Device type<br>4.Turn on Device Simulator and click simulation running. Enter the values of temperature, Humidity , Soil moisture<br>5.Click Send & Save. Verify the displayed result of the levels | Temperature, Humidity , Soil moisture sensor values are generated randomly in simulation |
| Node Red Installation | 1.Install node red and open node red in command prompt<br>2.Select IBM input in IoT | https://cloud.ibm.com/developer/appservice/create-app?starterKit=59c9d5bd-4d31-3611-897a-f94eea80dc9f&defaultLanguage=undefined |

| | | |
|---|---|---|
| Node Red Installation | 1.Select IBM IoT input in Node. In IBM IoT Watson Platform, go to apps and click on generate API keys.<br>2.Copy & paste generated API key and token in the IBM IoT input. After entering all details, click the done button.<br>3.Add debug to the IBM IoT and rename as Msg.payload and click on done. Click gauge from the dashboard and fill the details & add functions to the gauge. Check the generated values from the debug message.<br>4.Edit function node, connect them, add another gauge and functions, name them as "Temperature", "Soil moisture" &"Humidity"<br>5.Finally add light ON/OFF buttons to the IBM IoT and debug. Verify the output from NODE RED using Local host link | Values of sensors and button for light ON/OFF is displayed |
| Python 3.7.0(64 bit) installation | 1.Download and install Python 3.7.0<br>2.Develop python code | https://www.python.org/downloads/release/python-370/ |
| Python 3.7.0(64 bit) installation | 1.Downlinstall Python 3.7.0<br>2.After python code | Get the output from the code |
| IBM Cloud Login ID & Password | 1.Run the python code<br>2.Verify the displayed output | Publishment of python code |
| MIT Inventor Login ID & password | 1.Go to Node Red. Select http in & http response. Add functions and select another http in and http response. Connect them to IBM IoT output and function.Print the command statements such as light ON/OFF and sensor<br>2.Go to MIT app inventor and create frontend using buttons,horizontal arrangement, text bar, etc. Add blocks and so on to create back end. Verify the output | Sensors values and command values can be seen in the mobile application |
| IBM Cloud Login ID & Password | | Cloudant is connected by NODE RED |
| | | |

| | | |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

| Expected Result | Actual Result | Status | Comments |
|---|---|---|---|
| User should sign up IBM cloud and details should be verified | Working as expected | Pass | Results verified |
| User login to IBM Cloud and should be navigated to IBM Cloud dashboard page | Working as expected | Pass | Results verified |
| User should be navigated to IBM IoT Watson Platform | Working as expected | Pass | Results verified |
| Temperature, Humidity , Soil moisture sensor values should be randomly generated | Working as expected | Pass | Results verified |
| User should be able to see the Node Red page | Working as expected | Pass | Results verified |

| | | | |
|---|---|---|---|
| Values of sensors and button for light ON/OFF should be displayed | Working as expected | Pass | Results verified |
| User should be able to develop a python code | Working as expected | Pass | Results verified |
| User should be able to get the results from the developed code | Working as expected | Pass | Results verified |
| User should be able to publish the code | Working as expected | Pass | Results verified |
| Sensors values and command values should be seen in the mobile application | Working as expected | Pass | Results verified |
| User should be able to connect the Cloudant and Node Red | Working as expected | Pass | Results verified |
| | | | |

| | | | |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

| TC for Automation(Y/N) | BUG ID | Executed By |
| --- | --- | --- |
| No | | YAMUNA R,SHAFRIN K, SNEGA S,AMMU S |
| No | | YAMUNA R,SHAFRIN K, SNEGA S,AMMU S |
| No | | YAMUNA R,SHAFRIN K, SNEGA S,AMMU S |
| No | | YAMUNA R,SHAFRIN K, SNEGA S,AMMU S |
| No | | YAMUNA R,SHAFRIN K, SNEGA S,AMMU S |

| | | |
|---|---|---|
| No | | YAMUNA R,SHAFRIN K, SNEGA S,AMMU S |
| No | | YAMUNA R,SHAFRIN K, SNEGA S,AMMU S |
| No | | YAMUNA R,SHAFRIN K, SNEGA S,AMMU S |
| No | | YAMUNA R,SHAFRIN K, SNEGA S,AMMU S |
| No | | YAMUNA R,SHAFRIN K, SNEGA S,AMMU S |
| No | | YAMUNA R,SHAFRIN K, SNEGA S,AMMU S |
| | | |