

Project Development Phase

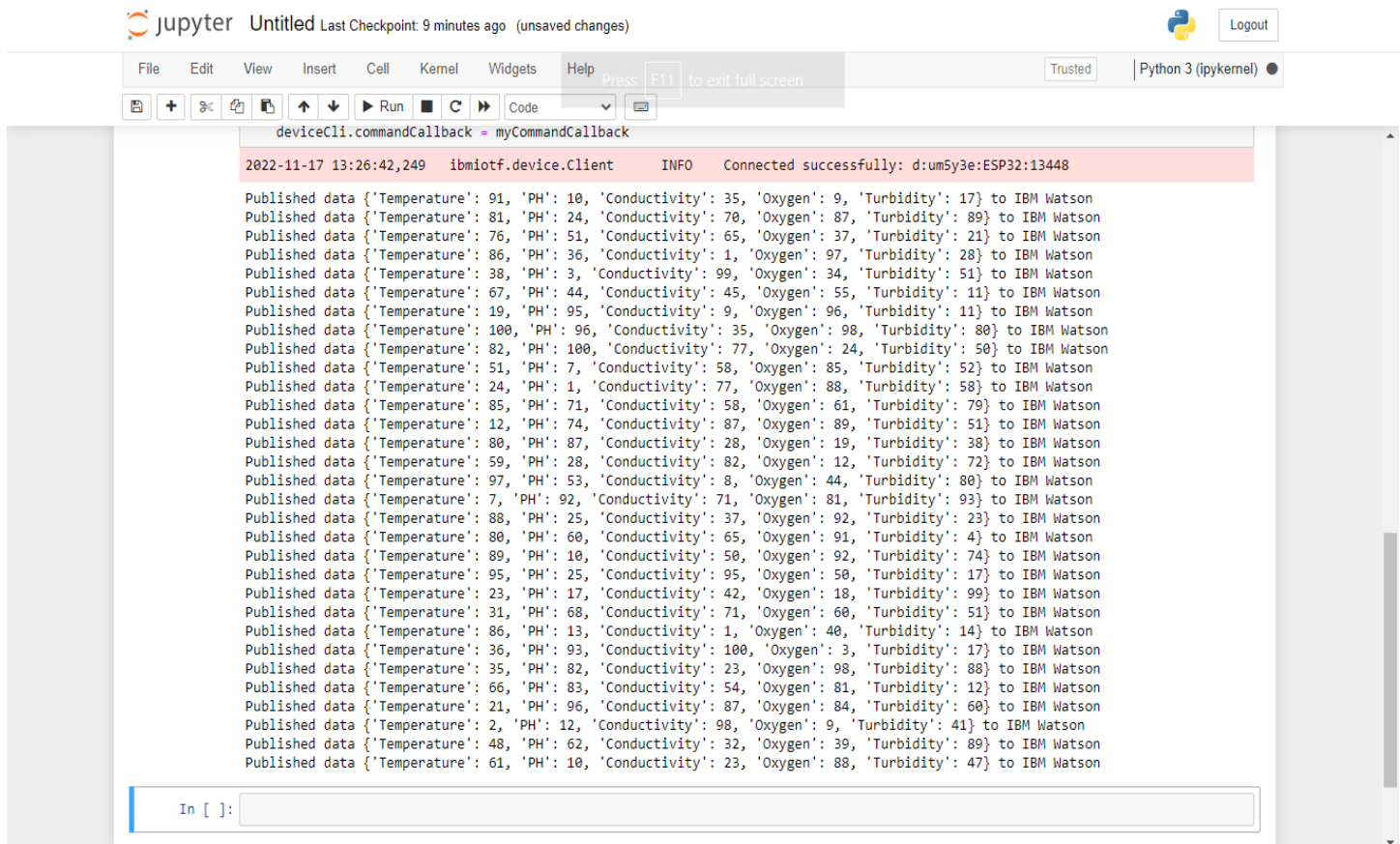
Sprint – 3

Date	13 Nov 2022
Team ID	PNT2022TMID06691
Project Name	Real-Time River Water Quality Monitoring and Control System
Maximum Marks	8 Marks

USN – 7: Publish Data to cloud

As a user, I can publish Data that is sensed by the microcontroller to the Cloud

Output in Python Shell (Jupyter Notebook):



The screenshot displays a Jupyter Notebook environment with the following components:

- Header:** "jupyter" logo, "Untitled" filename, "Last Checkpoint: 9 minutes ago (unsaved changes)", and a "Logout" button.
- Toolbar:** Includes "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help" menus. A "Press F11 to exit full screen" tooltip is visible. Below the menus are icons for file operations (save, copy, paste, etc.) and a "Run" button.
- Code Cell:** Contains the following output:

```
deviceCli.commandCallback = myCommandCallback
2022-11-17 13:26:42,249 ibmiotf.device.Client INFO Connected successfully: d:um5y3e:ESP32:13448
Published data {'Temperature': 91, 'PH': 10, 'Conductivity': 35, 'Oxygen': 9, 'Turbidity': 17} to IBM Watson
Published data {'Temperature': 81, 'PH': 24, 'Conductivity': 70, 'Oxygen': 87, 'Turbidity': 89} to IBM Watson
Published data {'Temperature': 76, 'PH': 51, 'Conductivity': 65, 'Oxygen': 37, 'Turbidity': 21} to IBM Watson
Published data {'Temperature': 86, 'PH': 36, 'Conductivity': 1, 'Oxygen': 97, 'Turbidity': 28} to IBM Watson
Published data {'Temperature': 38, 'PH': 3, 'Conductivity': 99, 'Oxygen': 34, 'Turbidity': 51} to IBM Watson
Published data {'Temperature': 67, 'PH': 44, 'Conductivity': 45, 'Oxygen': 55, 'Turbidity': 11} to IBM Watson
Published data {'Temperature': 19, 'PH': 95, 'Conductivity': 9, 'Oxygen': 96, 'Turbidity': 11} to IBM Watson
Published data {'Temperature': 100, 'PH': 96, 'Conductivity': 35, 'Oxygen': 98, 'Turbidity': 80} to IBM Watson
Published data {'Temperature': 82, 'PH': 100, 'Conductivity': 77, 'Oxygen': 24, 'Turbidity': 50} to IBM Watson
Published data {'Temperature': 51, 'PH': 7, 'Conductivity': 58, 'Oxygen': 85, 'Turbidity': 52} to IBM Watson
Published data {'Temperature': 24, 'PH': 1, 'Conductivity': 77, 'Oxygen': 88, 'Turbidity': 58} to IBM Watson
Published data {'Temperature': 85, 'PH': 71, 'Conductivity': 58, 'Oxygen': 61, 'Turbidity': 79} to IBM Watson
Published data {'Temperature': 12, 'PH': 74, 'Conductivity': 87, 'Oxygen': 89, 'Turbidity': 51} to IBM Watson
Published data {'Temperature': 80, 'PH': 87, 'Conductivity': 28, 'Oxygen': 19, 'Turbidity': 38} to IBM Watson
Published data {'Temperature': 59, 'PH': 28, 'Conductivity': 82, 'Oxygen': 12, 'Turbidity': 72} to IBM Watson
Published data {'Temperature': 97, 'PH': 53, 'Conductivity': 8, 'Oxygen': 44, 'Turbidity': 80} to IBM Watson
Published data {'Temperature': 7, 'PH': 92, 'Conductivity': 71, 'Oxygen': 81, 'Turbidity': 93} to IBM Watson
Published data {'Temperature': 88, 'PH': 25, 'Conductivity': 37, 'Oxygen': 92, 'Turbidity': 23} to IBM Watson
Published data {'Temperature': 80, 'PH': 60, 'Conductivity': 65, 'Oxygen': 91, 'Turbidity': 4} to IBM Watson
Published data {'Temperature': 89, 'PH': 10, 'Conductivity': 50, 'Oxygen': 92, 'Turbidity': 74} to IBM Watson
Published data {'Temperature': 95, 'PH': 25, 'Conductivity': 95, 'Oxygen': 50, 'Turbidity': 17} to IBM Watson
Published data {'Temperature': 23, 'PH': 17, 'Conductivity': 42, 'Oxygen': 18, 'Turbidity': 99} to IBM Watson
Published data {'Temperature': 31, 'PH': 68, 'Conductivity': 71, 'Oxygen': 60, 'Turbidity': 51} to IBM Watson
Published data {'Temperature': 86, 'PH': 13, 'Conductivity': 1, 'Oxygen': 40, 'Turbidity': 14} to IBM Watson
Published data {'Temperature': 36, 'PH': 93, 'Conductivity': 100, 'Oxygen': 3, 'Turbidity': 17} to IBM Watson
Published data {'Temperature': 35, 'PH': 82, 'Conductivity': 23, 'Oxygen': 98, 'Turbidity': 88} to IBM Watson
Published data {'Temperature': 66, 'PH': 83, 'Conductivity': 54, 'Oxygen': 81, 'Turbidity': 12} to IBM Watson
Published data {'Temperature': 21, 'PH': 96, 'Conductivity': 87, 'Oxygen': 84, 'Turbidity': 60} to IBM Watson
Published data {'Temperature': 2, 'PH': 12, 'Conductivity': 98, 'Oxygen': 9, 'Turbidity': 41} to IBM Watson
Published data {'Temperature': 48, 'PH': 62, 'Conductivity': 32, 'Oxygen': 39, 'Turbidity': 89} to IBM Watson
Published data {'Temperature': 61, 'PH': 10, 'Conductivity': 23, 'Oxygen': 88, 'Turbidity': 47} to IBM Watson
```
- Input Prompt:** At the bottom, there is an input prompt "In []:" followed by a text box.

Output in IBM Cloud:

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. The main content area shows details for a specific device (ID: 13448, ESP32, Connected). Below this, a 'Recent Events' tab is selected, displaying a table of live data streams. The table has columns for 'Event', 'Value', 'Format', and 'Last Received'. The data shows five recent events, each with a JSON payload containing Temperature, PH, and Conductivity values. The bottom of the screen shows a Windows taskbar with various application icons and system status information (86°F, Haze, 1:29 PM, 11/17/2022).

IBM Watson IoT Platform

61071912130@smartinternz.com
ID: um5y3e

Browse Action Device Types Interfaces Add Device +

13448 Connected ESP32 Device 16 Nov 2022 15:42

Identity Device Information Recent Events State Logs

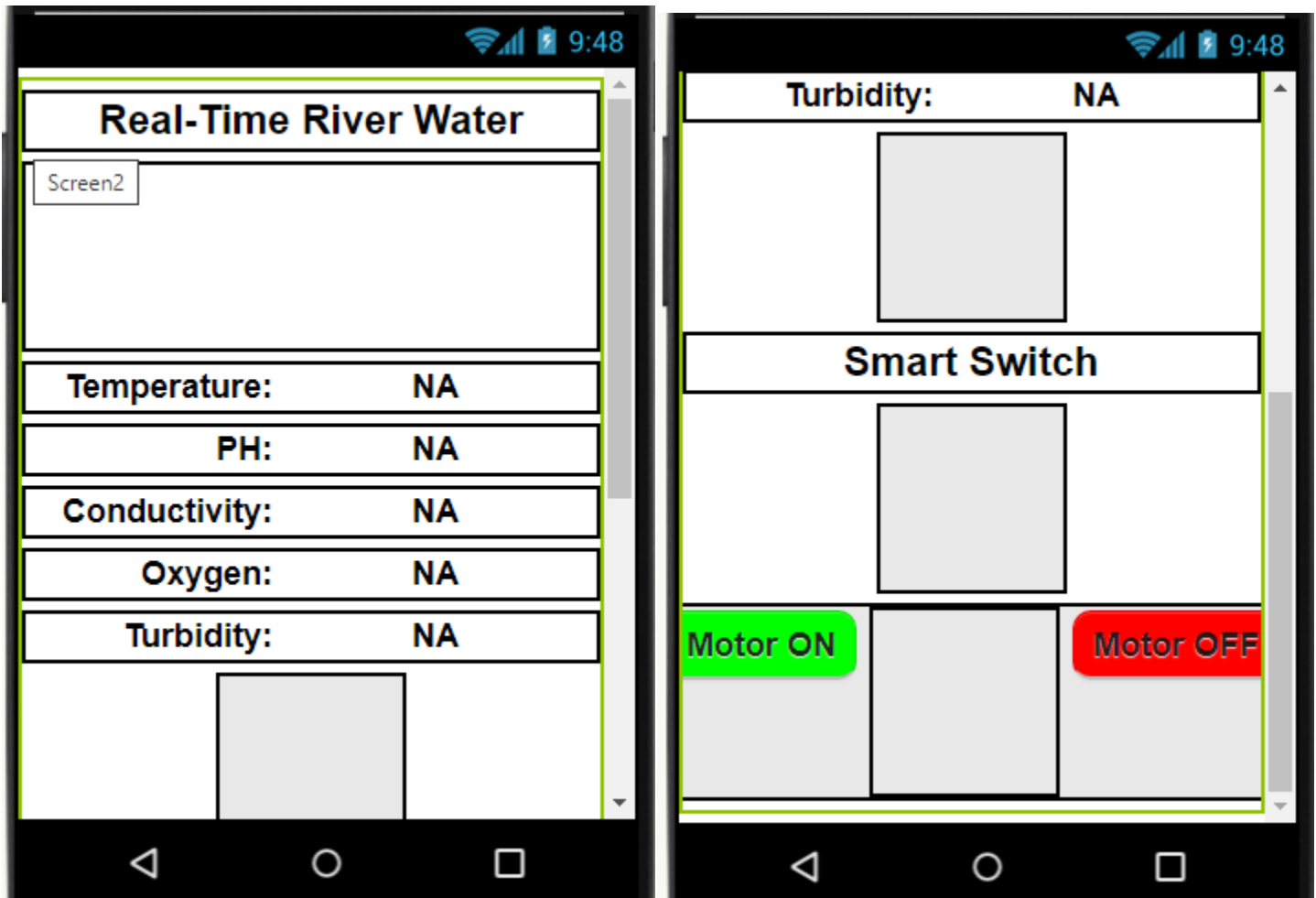
The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
event	{"Temperature":21,"PH":96,"Conductivity":87,"O...	json	a few seconds ago
event	{"Temperature":66,"PH":83,"Conductivity":54,"O...	json	a few seconds ago
event	{"Temperature":35,"PH":82,"Conductivity":23,"O...	json	a few seconds ago
event	{"Temperature":36,"PH":93,"Conductivity":100,"...	json	a few seconds ago
event	{"Temperature":86,"PH":13,"Conductivity":1,"Ox...	json	a few seconds ago

86°F Haze 1:29 PM 11/17/2022

USN – 8: MIT app inventor (Front end Design)

As a user, I can create the front end design for the application using MIT app Inventor



USN – 9: MIT app inventor (Back end Design)

As a user, I can create the back end design for the application using MIT app Inventor

