

Real-Time Weather based Smart Sprinkler System

Project title: “Real-Time Weather based Smart Sprinkler System”

Aim: To monitor sensor and weather data to control smart sprinkler system effectively on golf course.

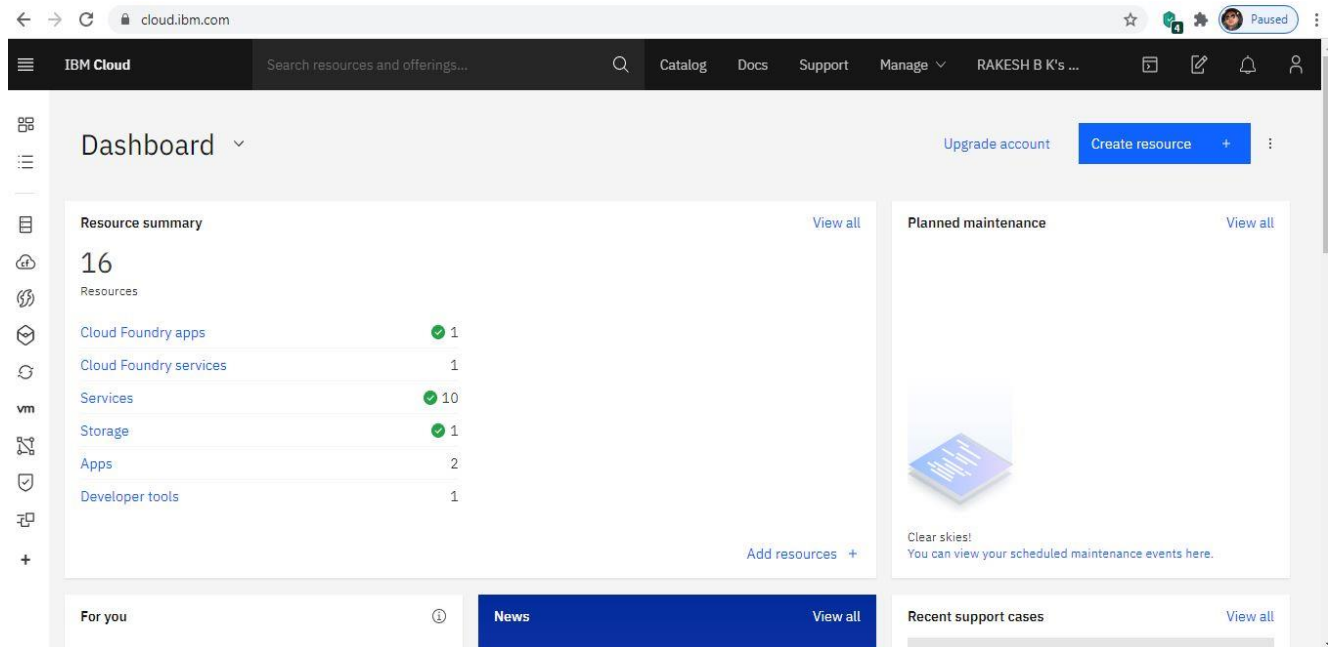
Category: Internet of Things

Skills Needed:

Service: IOT Cloud Platform ,Node- RED,IBM Cloudant DB, MIT app inventor to build mobile app.

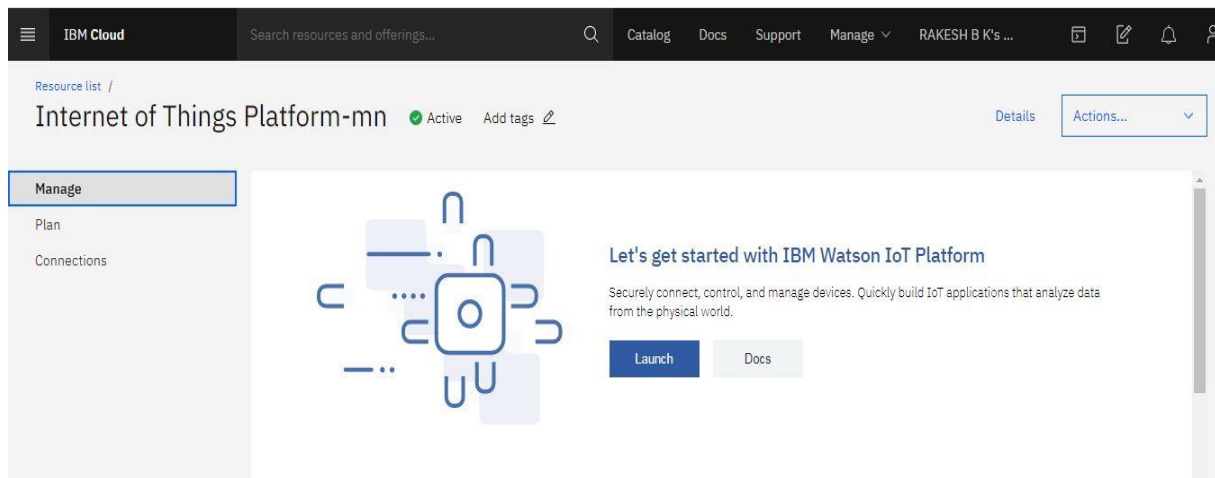
To begin with we need to create an account on IBM Cloud.com, select and create above mentioned services from catalog of IBM cloud dashboard menu along with their Api's.

IBM Cloud Dashboard



Select services from the catalog menu and create them with Api credentials.

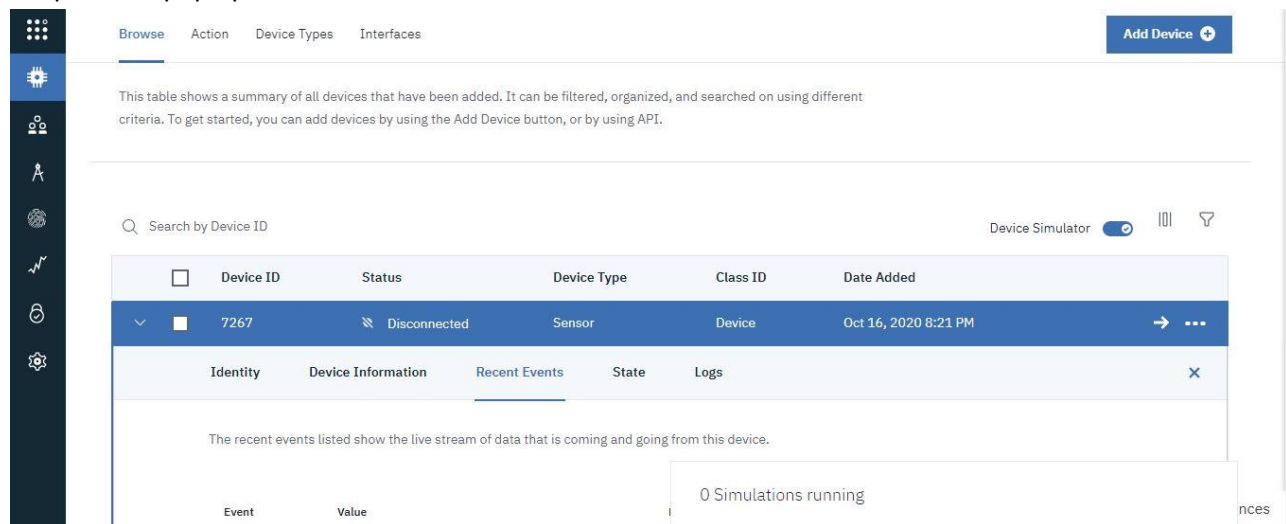
Once created launch the Service: IOT platform.



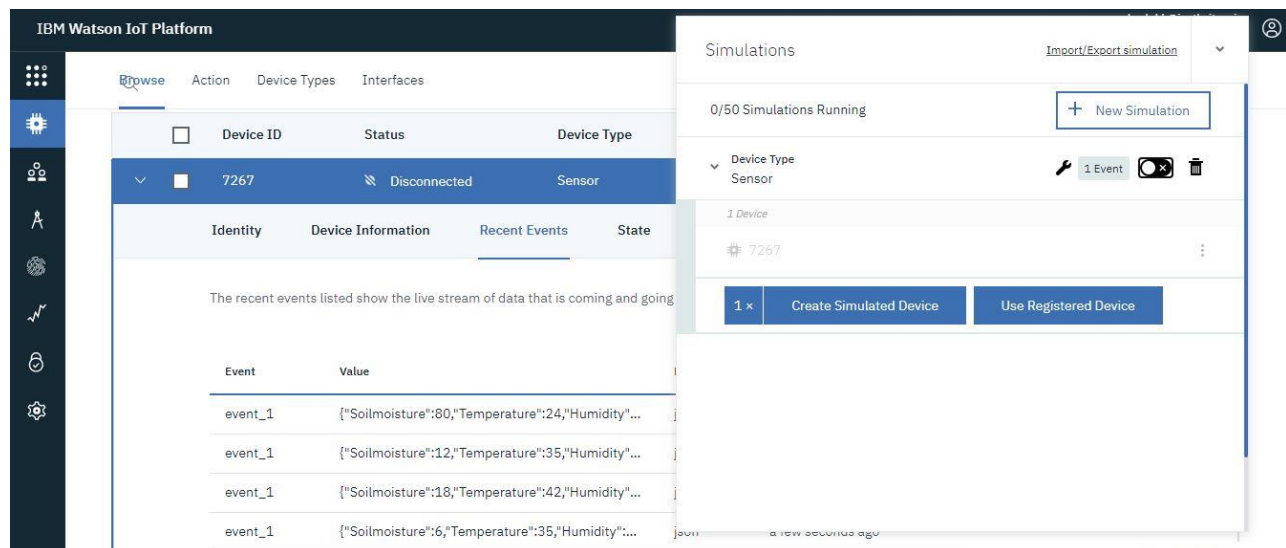
From IOT dashboard: Add device with system defined Api and Api token copy them on note pad for future usage. Follow the steps assigned during the process of creating IOT sensor device with respective credentials and enable device simulator on the menu bar to generate inbuilt sensor data through simulators.



As soon as you enable device simulator a pop window screen showing simulation running displays. Click on it to open new pop up window simulation.



Simulation window opens, select device type and device using registered device and enable button next to 1 Event and click on 1 Event.



Click on device type and create event, schedule the simulation for requirement and you can add payload for different sensors on the editor window as shown. If not upload CSV file directly of predefined sensor types.

The screenshot shows the IBM Watson IoT Platform interface. On the left, the 'Browse Devices' page displays a table with one device: ID 7267, Status 'Disconnected', and Device Type 'Sensor'. On the right, a modal window for 'Device Type: Sensor' is open, showing configuration options for an event named 'event_1'. The event is scheduled 'Every Minute' with a payload of random sensor data: 'Soilmoisture': random(0, 100), 'Temperature': random(0, 60), and 'Humidity': random(0, 100). A 'Send' button is visible to trigger the simulation.

Press the send button for simulation, the data can be viewed on recent events from the device tab as shown in the previous picture as event_1...

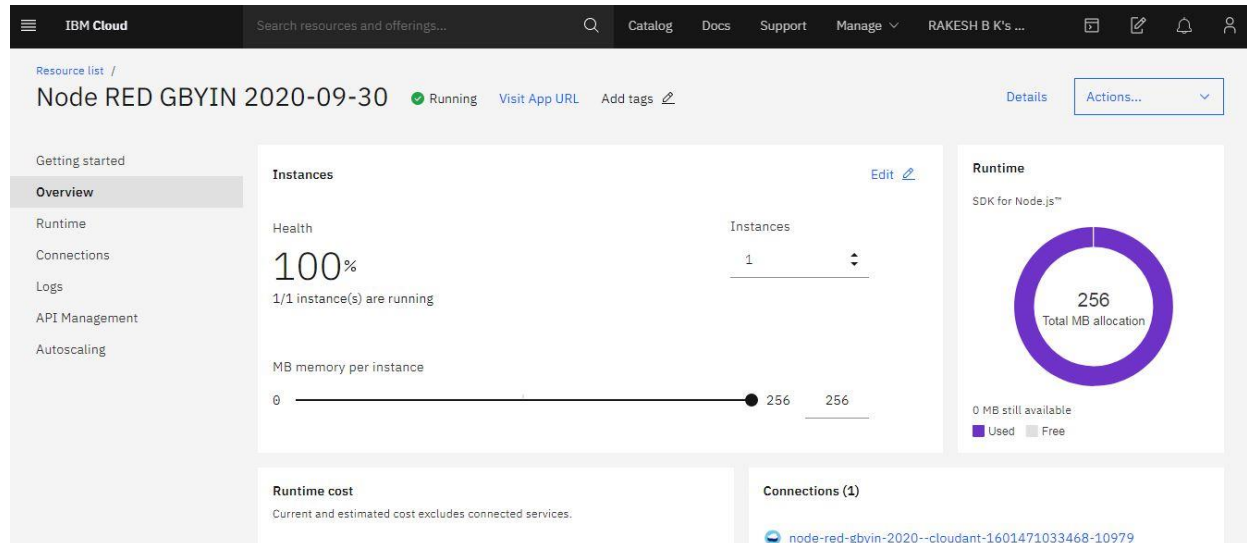
This simulated data is automatically stored on IBM cloudant by creating node-red cloudant service from the IBM cloud dashboard. Click on launch dashboard.

The screenshot shows the IBM Cloud dashboard for a service named 'node-red-gbyin-2020--cloudant-1601471033468'. The 'Overview' tab is selected, displaying deployment details: CRN, Location (Sydney), External Endpoint, and Authentication methods (IBM Cloud IAM and Cloudant credentials). A 'Launch Dashboard' button is present in the top right.

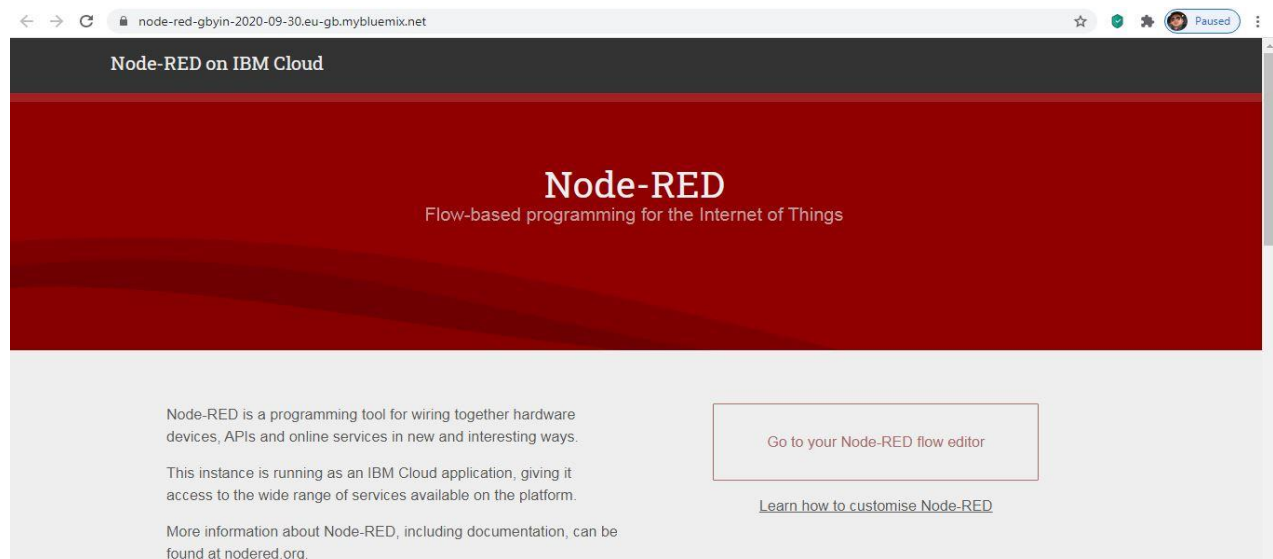
Database window opens on new tab, use create database to name a new storage space on IBM cloud.

The screenshot shows the 'Databases' window in the IBM Cloud interface. It features a table listing three databases: 'iotdata' (54.0 KB, 140 docs), 'nodered' (126.1 KB, 4 docs), and 'sensordata' (41.9 KB, 92 docs). Each database has icons for actions like refresh, delete, and export. A 'Create Database' button is located at the top right.

After the above step, go to IBM cloud dashboard to create a service node-red application from cloud foundry app. Redirected to below page, click on visit app URL to start with new window Node-red on IBM cloud.



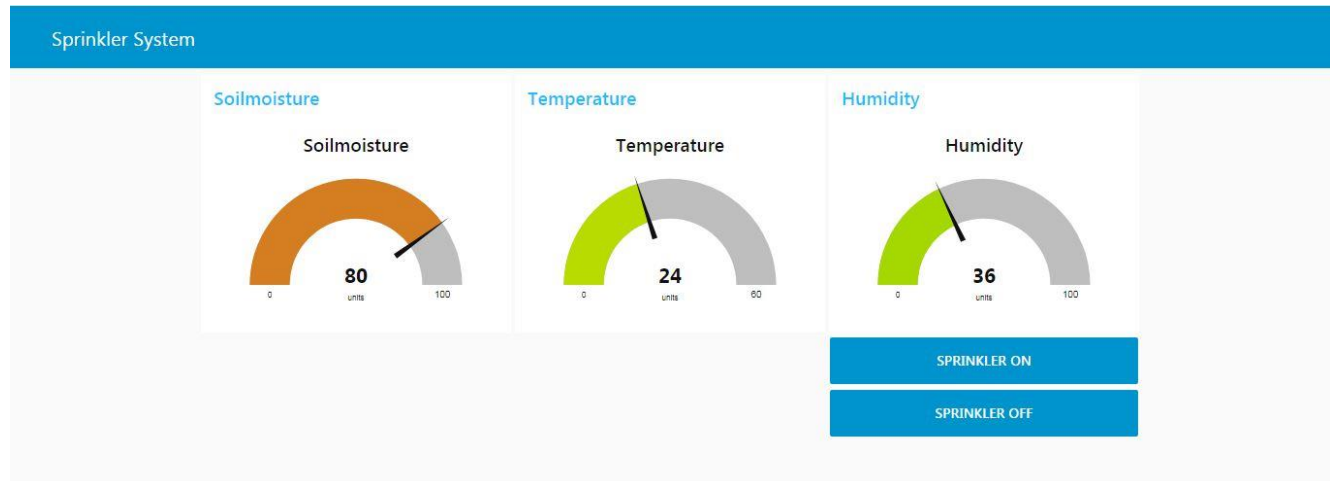
Click on go to Node-RED flow editor



Create a flow name as required show in the figure. Using respective nodes build a flow to generate UI for web, to store data on cloudant, to fetch data from cloud to web, mobile app UI using HTTP nodes.



Go to right top corner of previous picture to access node-red dashboard and click on the pop up button next to theme, redirected to new tab as displayed below or use this URL (<https://node-red-gbyin-2020-09-30.eu-gb.mybluemix.net/ui>) to visualize the simulated sensor data on web UI



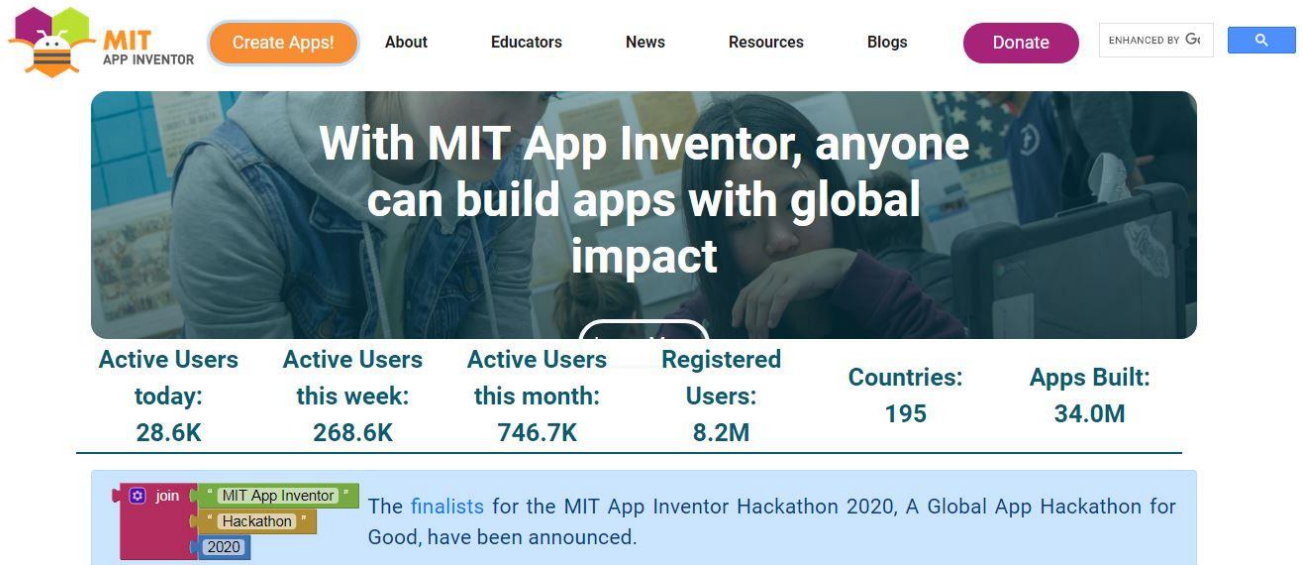
To access real-time weather data create an account in Open Weather Map website. Get provided with unique API. Go to current weather data API Doc to find API syntax.

The screenshot shows the OpenWeather website's 'Weather API' page. It includes a navigation bar with links like 'Get Started', 'API', 'Pricing', 'Maps', 'FAQ', 'Partners', 'Blog', 'Marketplace', and 'Support'. The main content area features a heading 'Weather API' and a section titled 'Current & Forecast weather data collection' with three cards: 'Current Weather Data', 'Hourly Forecast 4 days', and 'One Call API NEW'. Each card provides a brief description of the API and a 'Subscribe' button.

Select API call, first syntax representing “city name” copy this syntax, provide your city name whose weather data to be forecasted and use generated API from your account replacing API key on the syntax and paste it to http request node URL to access, monitor current weather data on web UI and on mobile app UI.

The screenshot shows the OpenWeather website's 'Call current weather data for one location' page. It includes a navigation bar with links like 'Get Started', 'API', 'Pricing', 'Maps', 'FAQ', 'Partners', 'Blog', 'Marketplace', and 'Support'. The main content area features a heading 'Call current weather data for one location' and a section 'By city name' with a paragraph about calling by city name or city name, state code and country code. Below this, there are two API call examples: 'api.openweathermap.org/data/2.5/weather?q={city name}&appid={API key}' and 'api.openweathermap.org/data/2.5/weather?q={city name},{state code}&appid={API key}'. To the right of the main content, there is a sidebar with a list of features and API options.

Create a MIT app Inventor account using appinventor.mit.edu web portal to build a mobile app. Click on create apps on the menu.

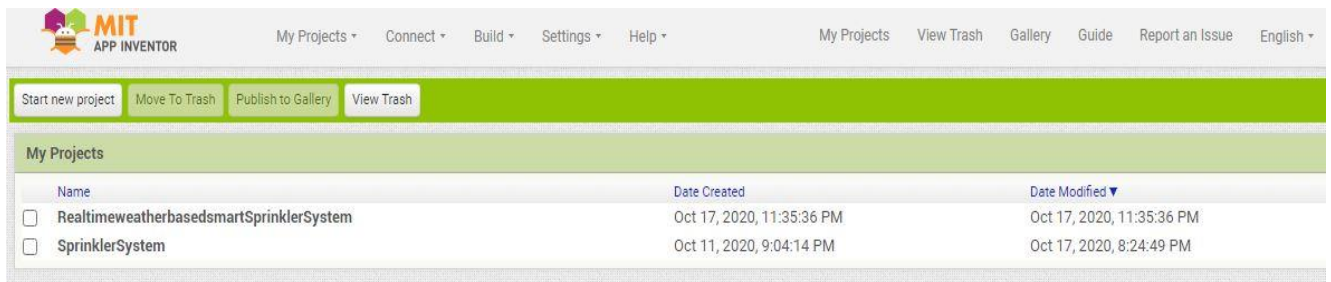


The MIT App Inventor homepage features a banner with the text "With MIT App Inventor, anyone can build apps with global impact" over a background image of students. Below the banner is a statistics table and a news section.

Active Users today:	Active Users this week:	Active Users this month:	Registered Users:	Countries:	Apps Built:
28.6K	268.6K	746.7K	8.2M	195	34.0M

The finalists for the MIT App Inventor Hackathon 2020, A Global App Hackathon for Good, have been announced.

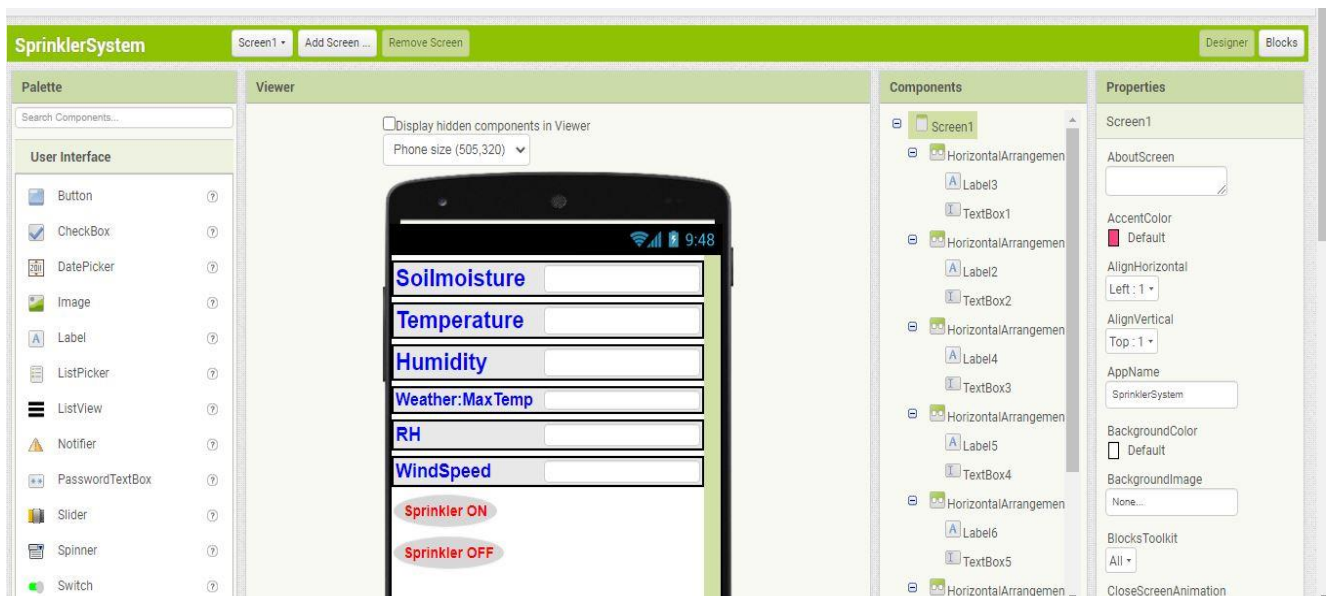
Get redirected to MIT app inventor dashboard. Click on start a new project and name the project.



The MIT App Inventor dashboard shows a navigation bar with options like "My Projects", "Connect", "Build", "Settings", and "Help". Below the navigation bar, there are buttons for "Start new project", "Move To Trash", "Publish to Gallery", and "View Trash". The "My Projects" section displays a table of projects.

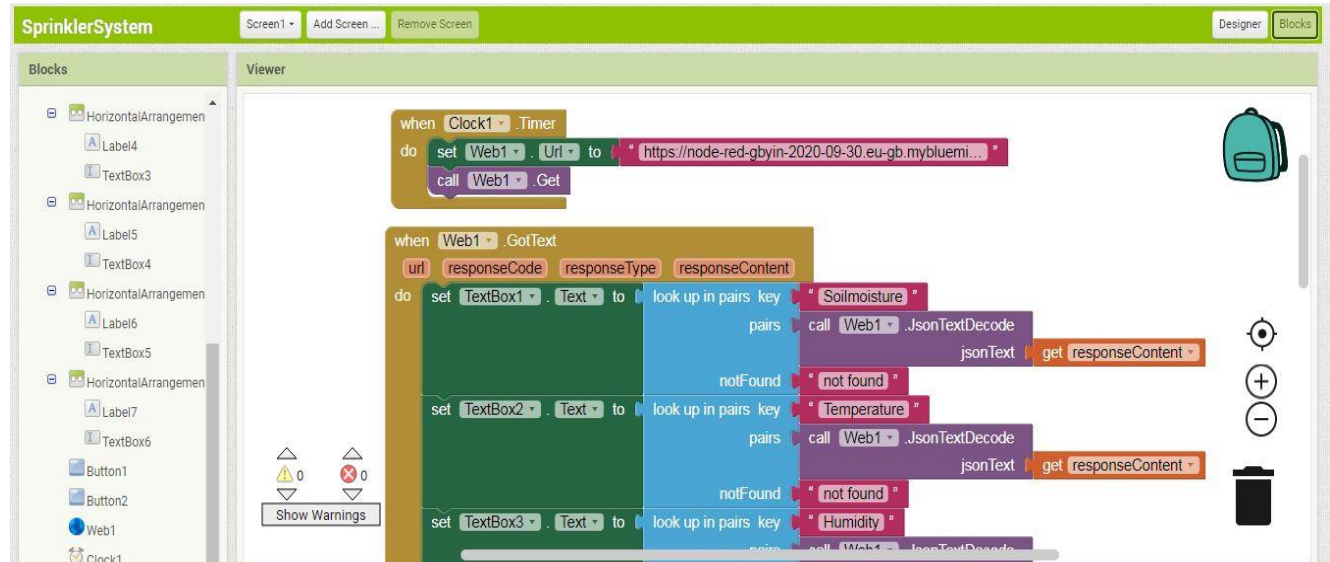
Name	Date Created	Date Modified
<input type="checkbox"/> RealtimeweatherbasedsmartSprinklerSystem	Oct 17, 2020, 11:35:36 PM	Oct 17, 2020, 11:35:36 PM
<input type="checkbox"/> SprinklerSystem	Oct 11, 2020, 9:04:14 PM	Oct 17, 2020, 8:24:49 PM

Get redirected to below screen showing mobile virtual screen called as designer. Start designing an UI for front end of the app using palettes on left corner of the screen as drop down menu as shown in the below picture.



The MIT App Inventor Designer interface shows a mobile virtual screen for a project named "SprinklerSystem". The screen displays a list of weather-related data: Soilmoisture, Temperature, Humidity, Weather: MaxTemp, RH, and WindSpeed. Below the list are two buttons: "Sprinkler ON" and "Sprinkler OFF". The interface includes a Palette on the left with various UI components, a Viewer in the center showing the mobile screen, a Components panel on the right listing the components on the screen, and a Properties panel on the far right for configuring the selected component.

Once the mobile UI front end been constructed. Click on blocks button available on the right top corner of the below picture. Build back end using respective blocks to request sensor data from the cloudant and real time weather data from open weather map using http request on node-red flow editor.



Create two separate sprinkler button to ON/OFF the smart sprinkler system as show using mobile building blocks.

