BACH KHOA UNIVERSITY
FALCULTY OF COMPUTER SCIENCE AND ENGINEERING

AUTOMATIC WATERING

[IoT group's project]

Group members:
Nguyen Quang Cuong - 1410420
Nguyen Thanh Van - 1450395
Dinh Gia Bao - 1450010
Nguyen Hien Quang - 1450124
Mai Hoang Phi - 1552276
Tran Duc Trung - 1552402

Table of Contents

I. INTRODUCTION

1. INTRODUCTION OF GROUP IDEA

Viet Nam is one of country which is developed base on agriculture, such as: export café and rice. For the convenience and best result at the end of season, watering and measurement in agriculture are the most important aspect. Due to the demand and increase the efficiency of the product, our group decided to apply our class subject-IoT –to help farmers have best qualified products.


IoTs – Internet of Things- is a system which connect hardware and software to improve human life. Not the connection only between hardware and software, IoTs help people easy to use and convenience in create or upgrade their product with low cost and high quality.

Our project's name is Automatic Watering. This project is based on Android Things to measure the temperature and humidity, which is collected from the garden outside. These data will be pushed into the Server which is created by our group for storing and display in the UI.

2. REASON

The project components:

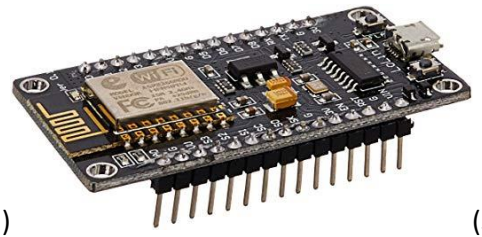
a. Hardware

DHT-11: Collect humidity and temperature in the garden. (1)

Node MCU esp8266: Connect to Wi-Fi and transmit data, which is collected from sensor DHT-11, to Server for UI display. (2)

Raspberry pi 3 model B: Control the sensor condition and number of node connect to the system. If the data value is higher than the condition, led will blink. (3)

LED: Blink when the data value is higher than the condition.

(1)  (2)  (3) 

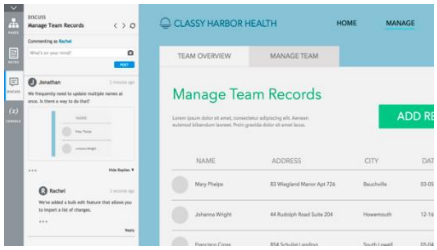
b. Software

OS: Android Things, easy for user to use and we are learning in class. This OS is developed by Google and has same function as android mobile.

Server:

- Language of back-end: Javascript (Nodejs platform)
- Database: MongoDB (Mongoose integrated)
- Server framework: Expressjs
- Cloud platform server: Heroku
- Language in front-end for getting data: php, Javascript,…

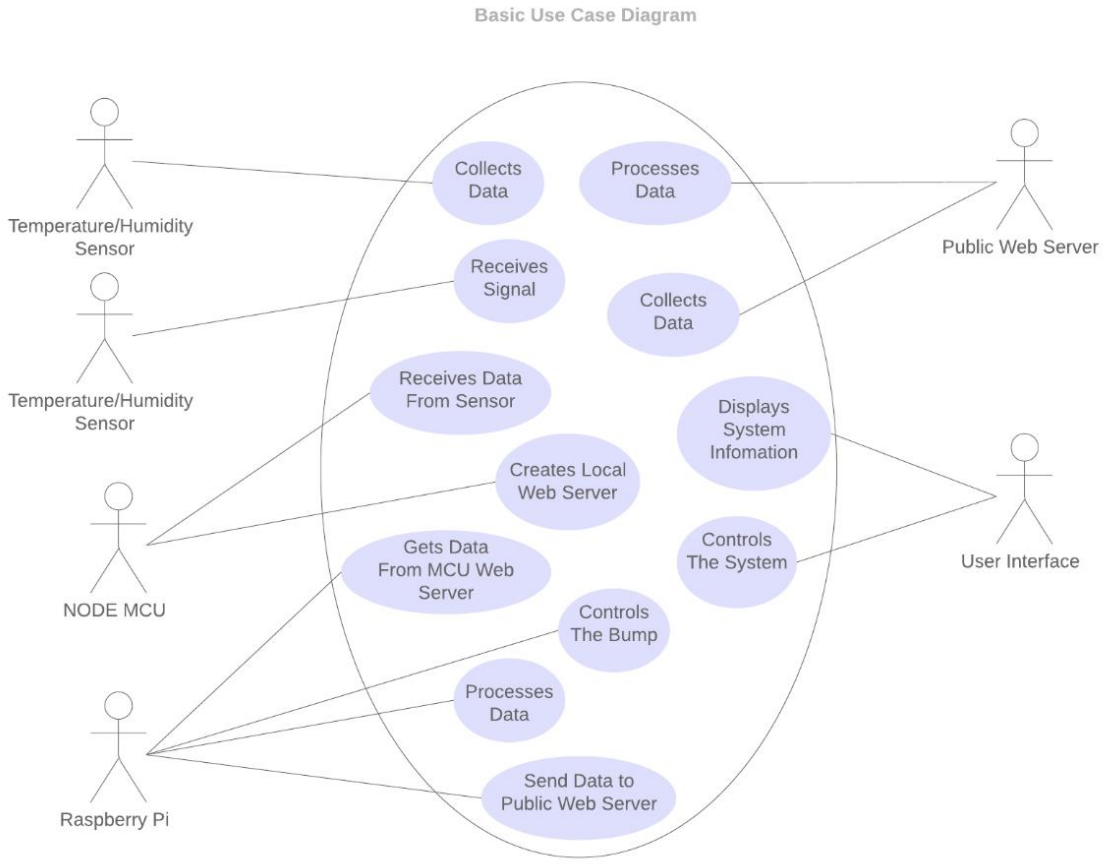
Website: Axure RP 8 is one of most famous application for designing website UI. It is easy to share, diagraming, documentation and easier team collaboration.

(1) 

(2) 

II. DESIGN

Use case diagram



- Sensors have to collect data from the environment then push it to node MCU
- Node MCU collect data from sensors, then create a local server for transferring data to the gateway.
- Raspberry Pi gets data from node MCU server. After that, the data is formatted and push to the Web server. If the data is higher than the condition, which is set on the server by user, Raspberry Pi will control the bump to bump water if the plan needs water.
- Public web server will receive data, which is sent by the Pi, then display it to the UI.
- UI will display value for user easy to see the temperature and humidity; see current connecting node MCU. Moreover, user can adjust the condition then send back to the Pi.

III. IMPLEMENTATION

1. Hardware

- Connect DHT-11 to node MCU.
- Run Raspberry Pi: connect to Wi-fi, Server and nod MCU.
- Setup node MCU esp8266: connect to Wi-Fi and raspberry.

2. Server

Detail:

- Create a skeleton for back-end (language, database type, database structure, local server…).
- Write API for retrieving specify data.
- Deploy back-end into a host site (includes site for api and database online) so it can be accessed easily for testing purpose.
- Write code in front-end pages to get data from server.

Database structure (tables): At the moment, our subject has 2 tables in the database:

- Condition: {

temp_condition:Number,

humid_condition:Number,

mcu_id:Number,

mode:String,

timestamp: Date

}

where

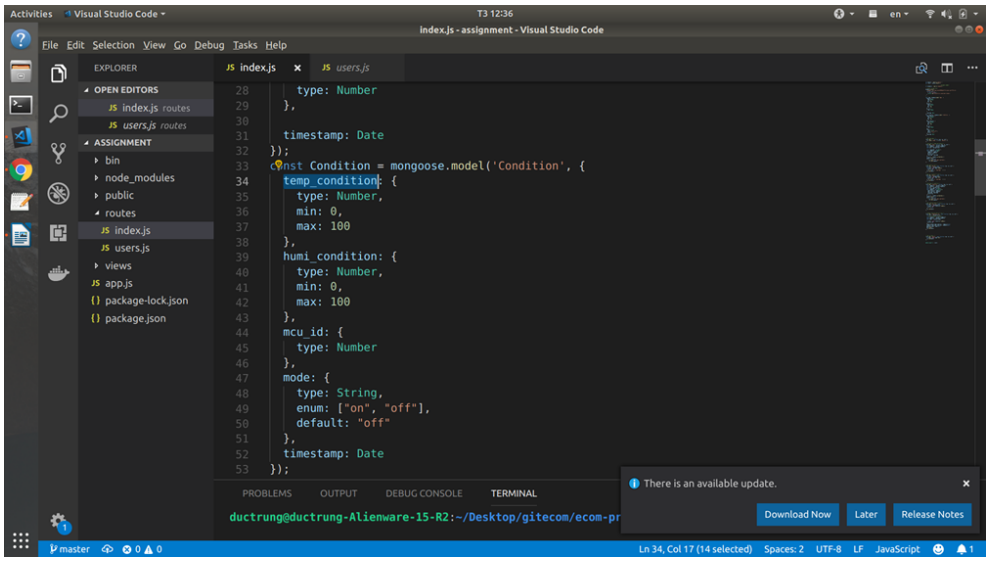
temp_condition is the condition temporary for bumping water.

humid_condition is the condition humidity for bumping water.

mcu_id is uid of specify mcu.

mode is status of ON or OFF of bumping machine.

timestamp is the time that user set or change condition.



- App: {

    temporary:Number,

    humidity:Number,

    mcu_id:Number,

    timestamp: Date

}

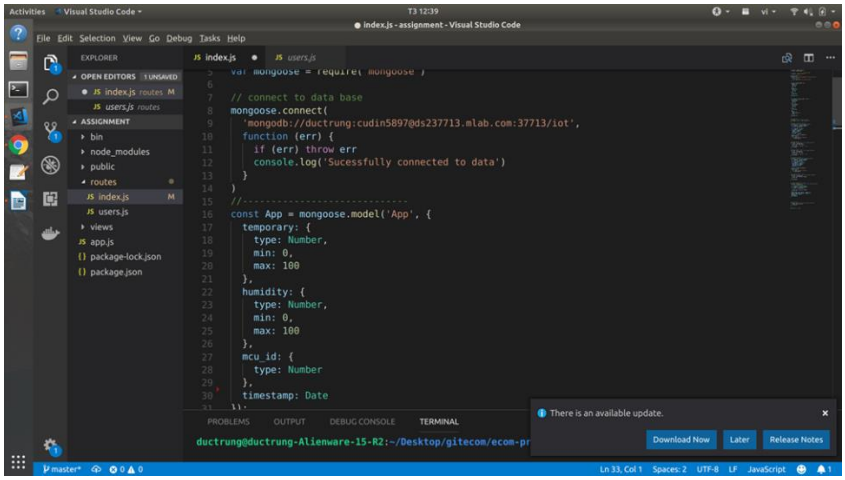
where

    temporary is the current temporary of ground.

    humidity is the current humidity of ground.

    mcu_id is uid of specify mcu.

    timestamp is the time the machine send data.

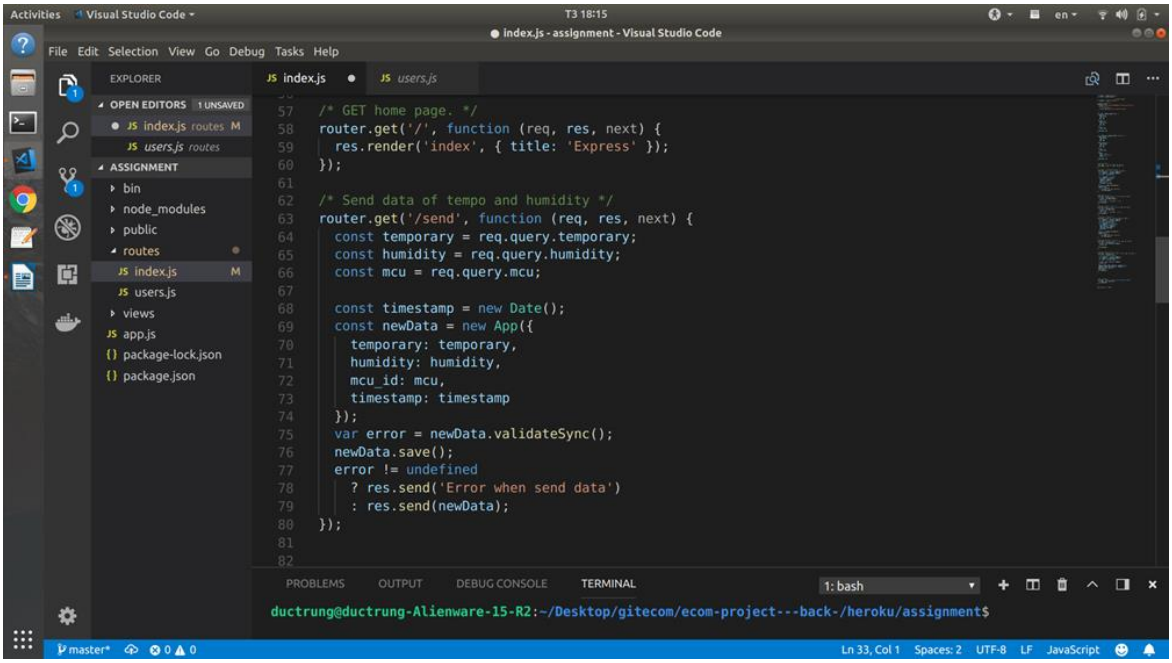


2) Write API:

In this project, rashpberry pi will receive and send data to server.

- Write api for sending data to database: which requests user to input current temporary, humidity, mcu_id. Example for this api:

https://iot-assignment.herokuapp.com/send?temporary=30&humidity=60&uid=3488A17988A139

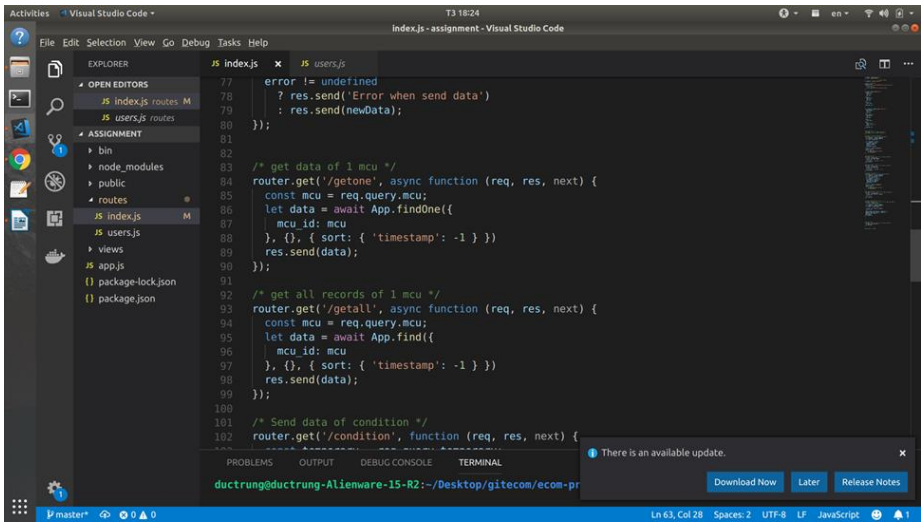


- Api for getting the latest data of mcu due to its uid which request uid of mcu

Example for this api: https://iot-assignment.herokuapp.com/getone?mcu=3488A17988A139

- And another api for getting all data of that mcu:

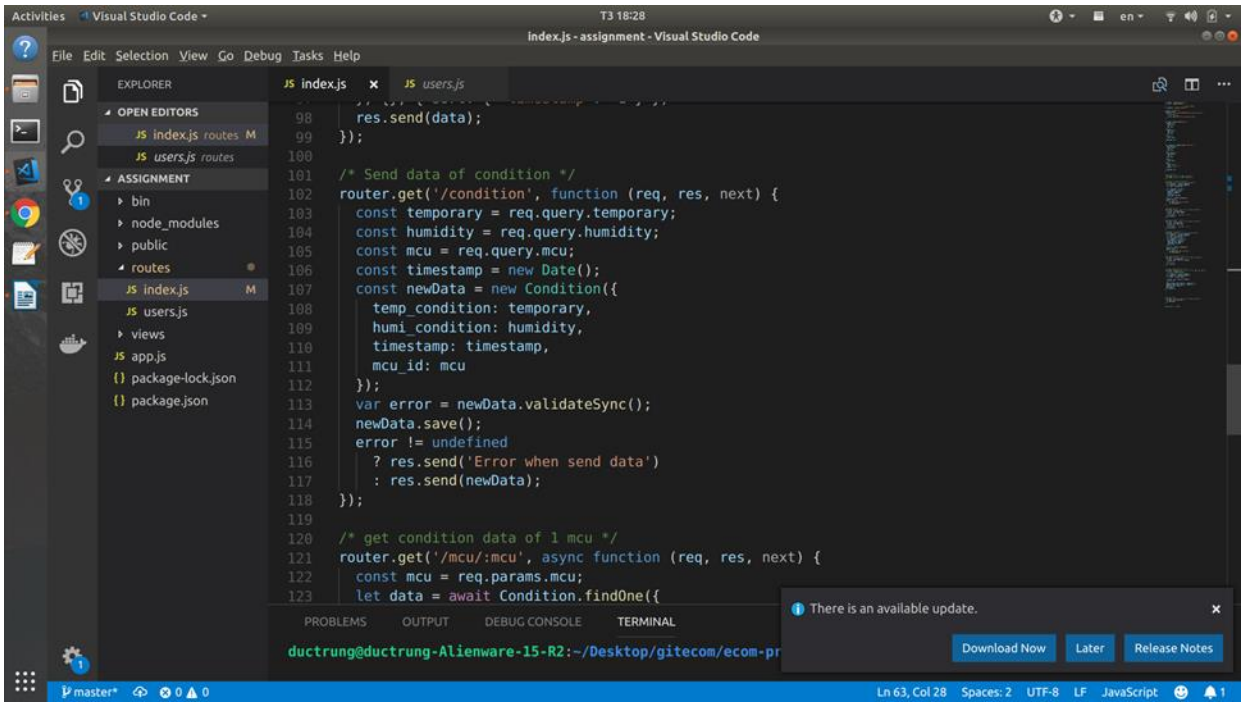
Example for this api: https://iot-assignment.herokuapp.com/getall?mcu=123



- Api to set the condition of temporary and humidity for the bump machine to water.

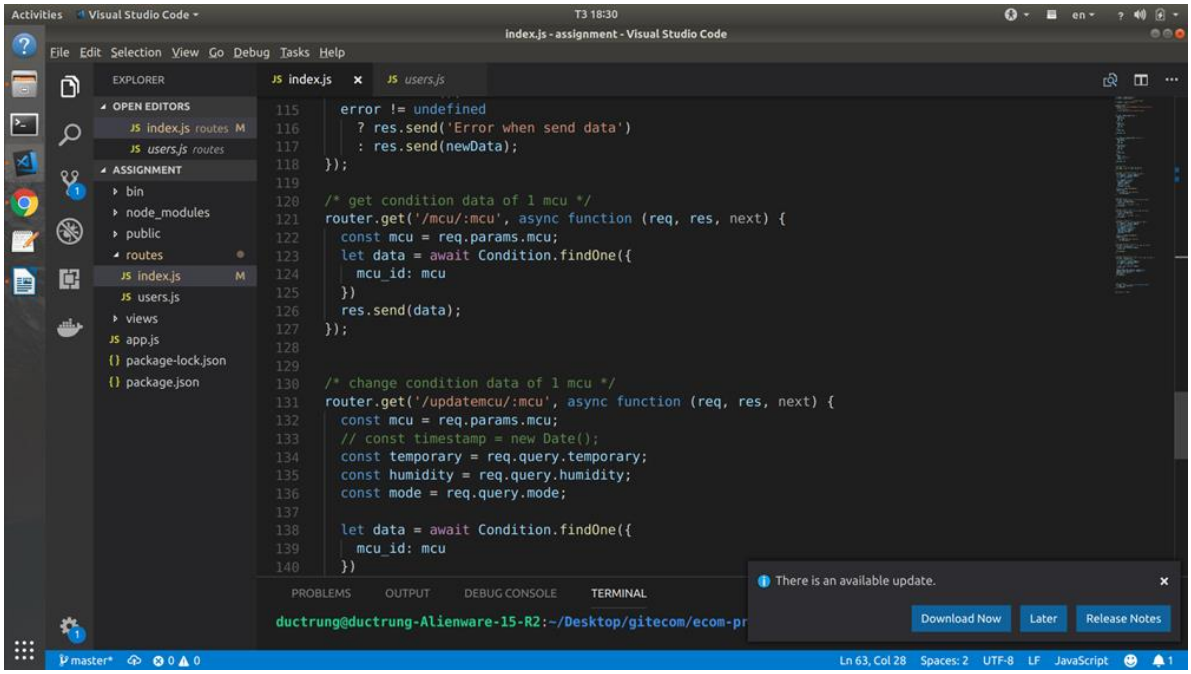
It requests temporary condition, humidity condition, mcu id. Example for this api:

https://iot-assignment.herokuapp.com/condition?temporary=28&humidity=58&mcu=3488A17988A139

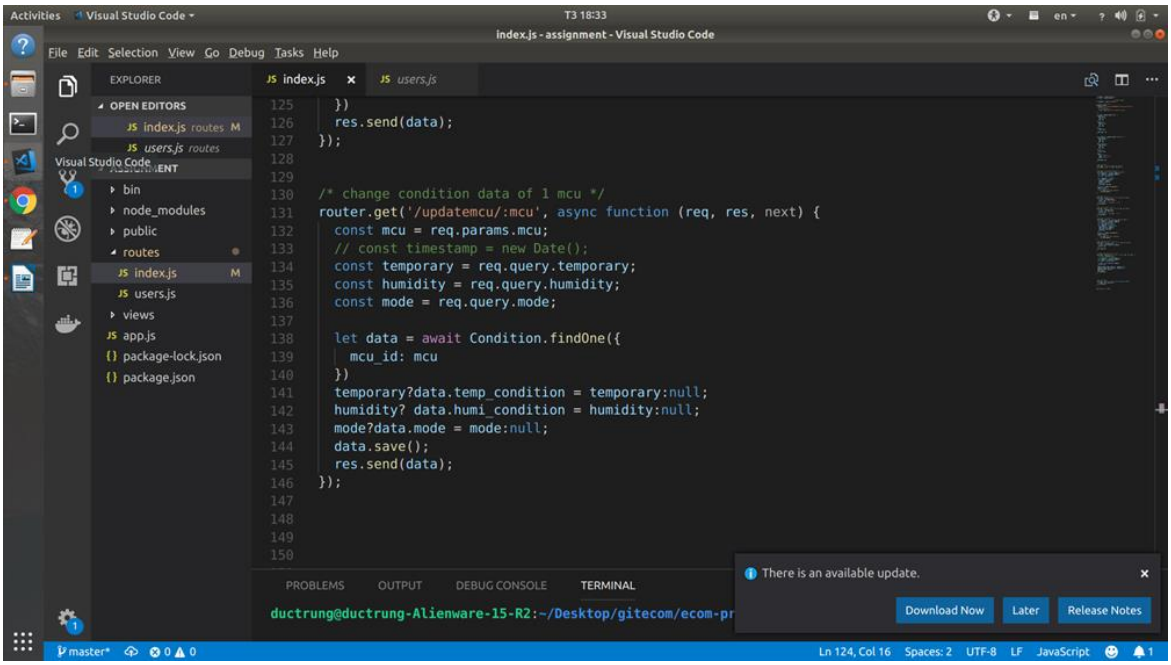


- Api to get condition of one mcu which requests mcu uid. Example for this api:

https://iot-assignment.herokuapp.com/mcu/123



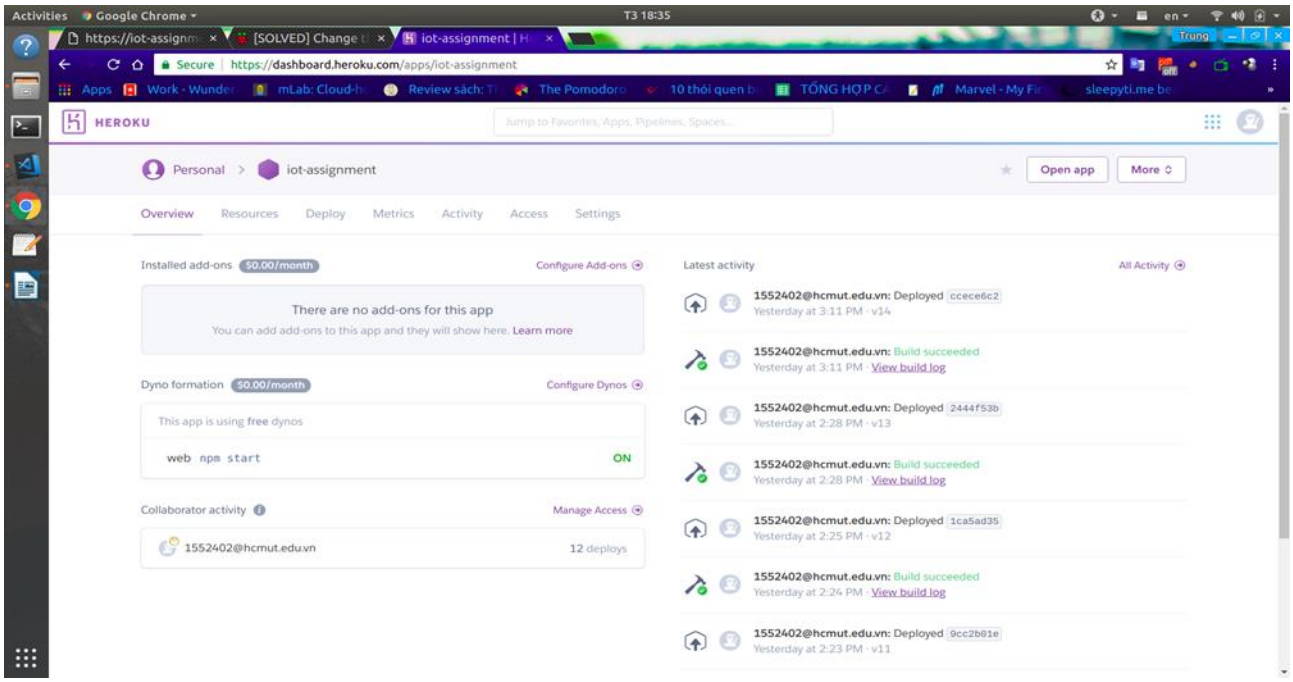
- Api to change condition of one mcu which requests mcu uid, humidity condition, temporary condition. Example for this api:

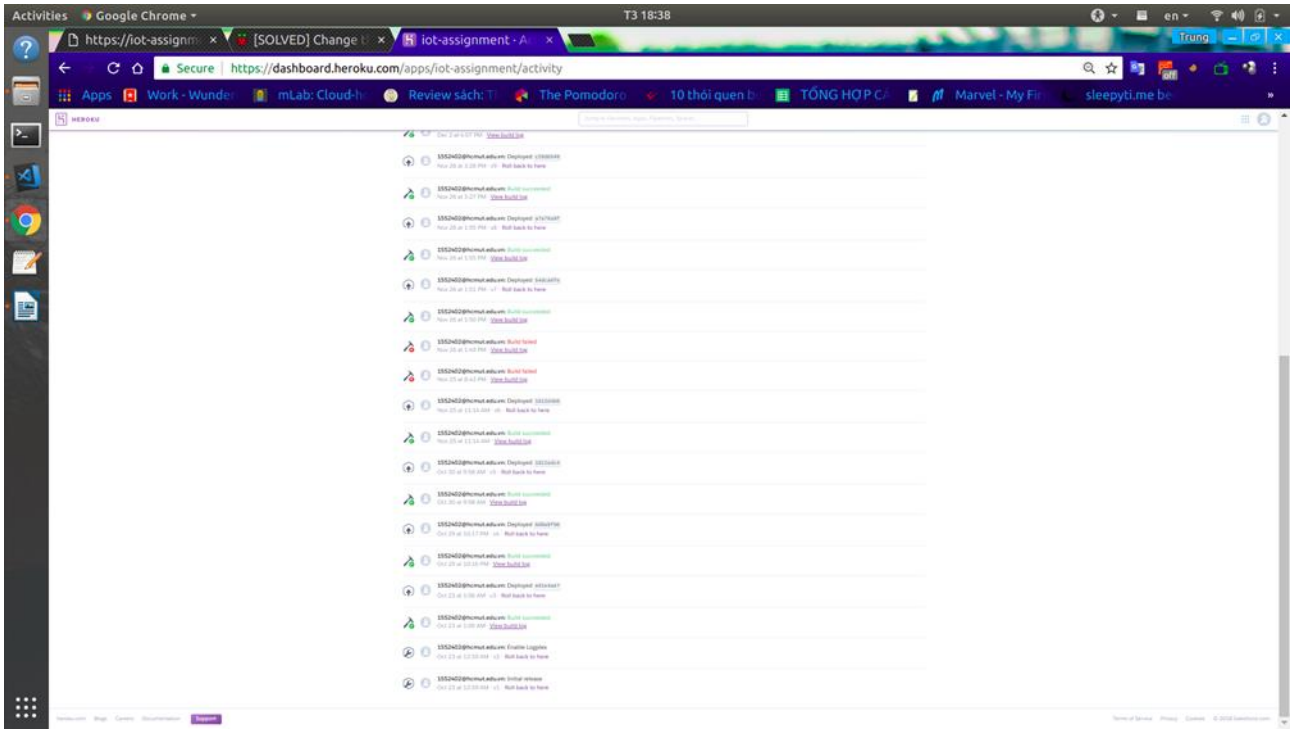


https://iot-assignment.herokuapp.com/updatemcu/123?temporary=27&humidity=80

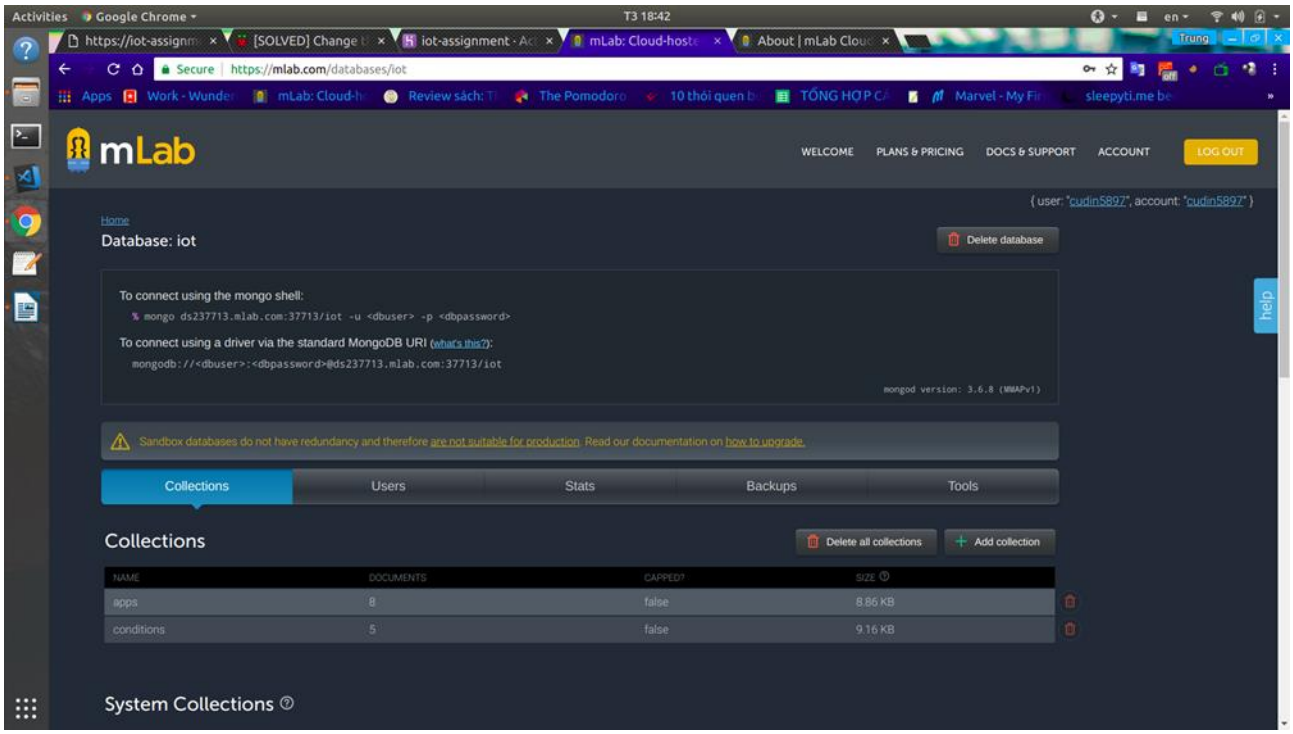
3) Deploy back-end into a host site (api and database online)

- Heroku is free host site and its can be integrated with Nodejs so I can use it to update the apis which created in local and push them by GIT.





- For the database I use a free cloud database name MLAB which give me 500mb for storage.

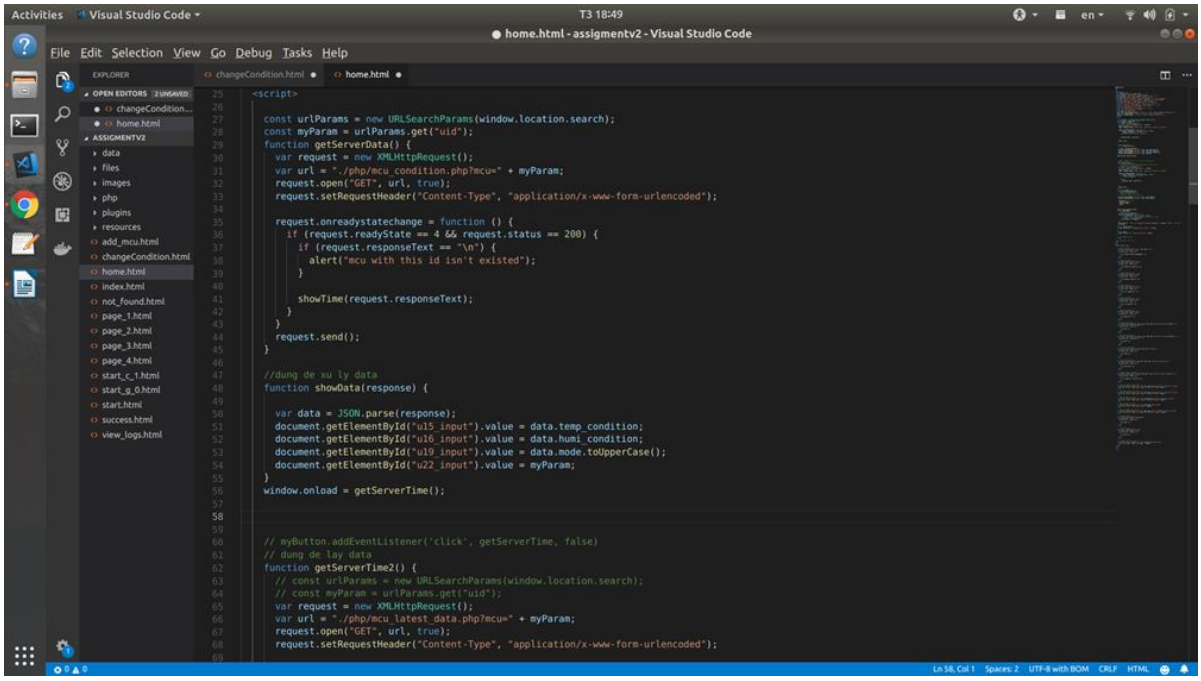


4) Write code in front-end pages to get data from server.

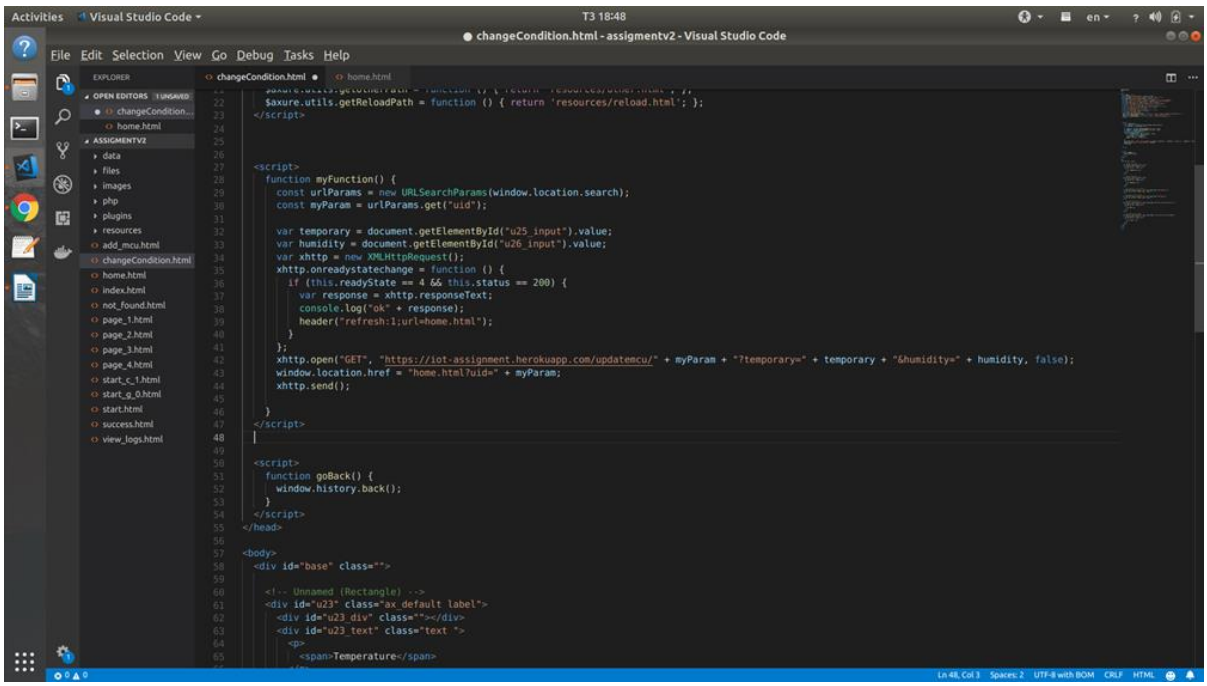
In this part, I usually use javascript in html page and sometimes use php for retrieving data from api (because of something ‘s not working when retrieving directly from api).

Here are some examples of functions using to access api:

Function to get condition:

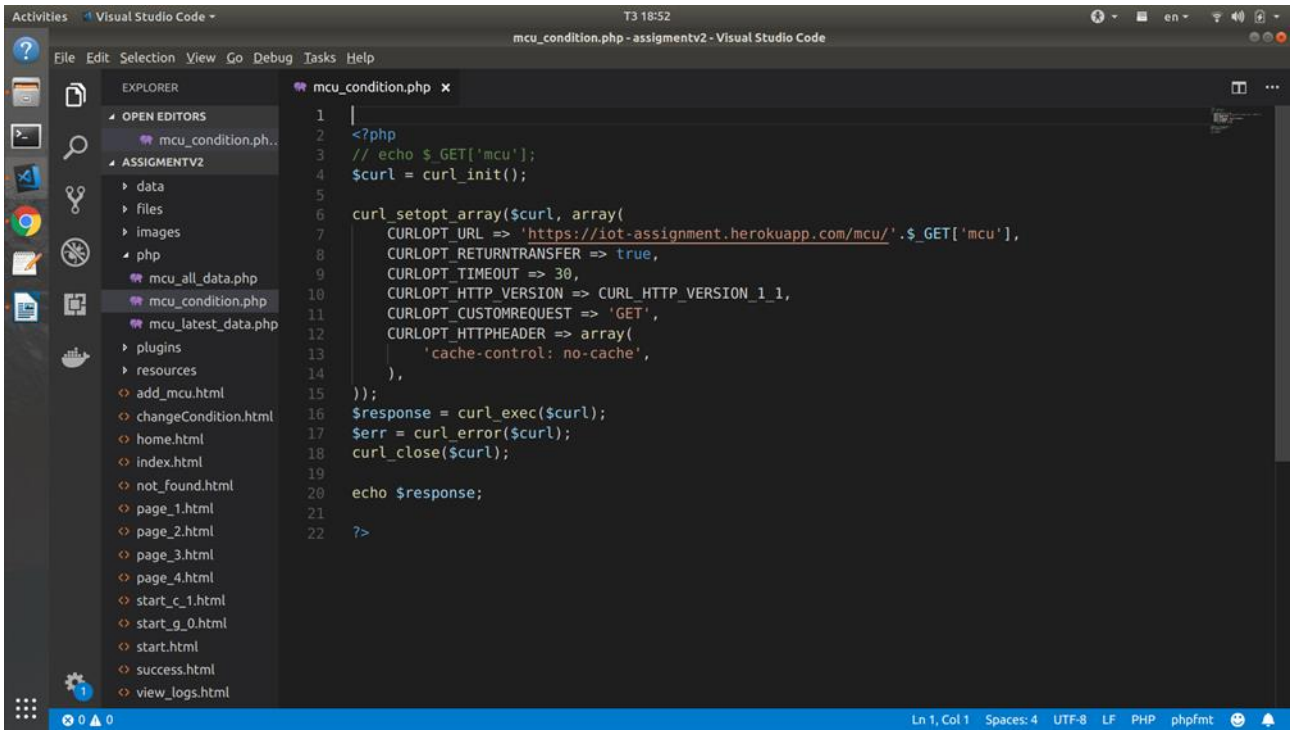


Function to update condition:

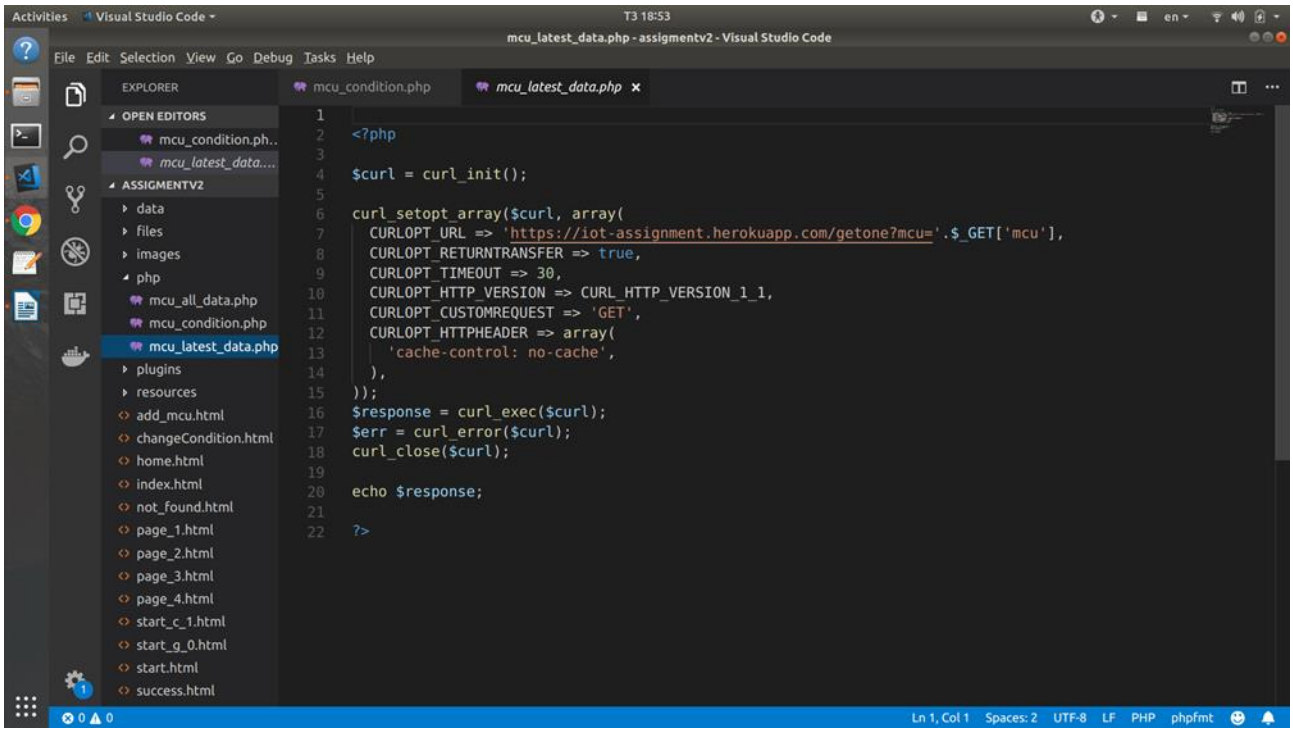


PHP file for accessing api:

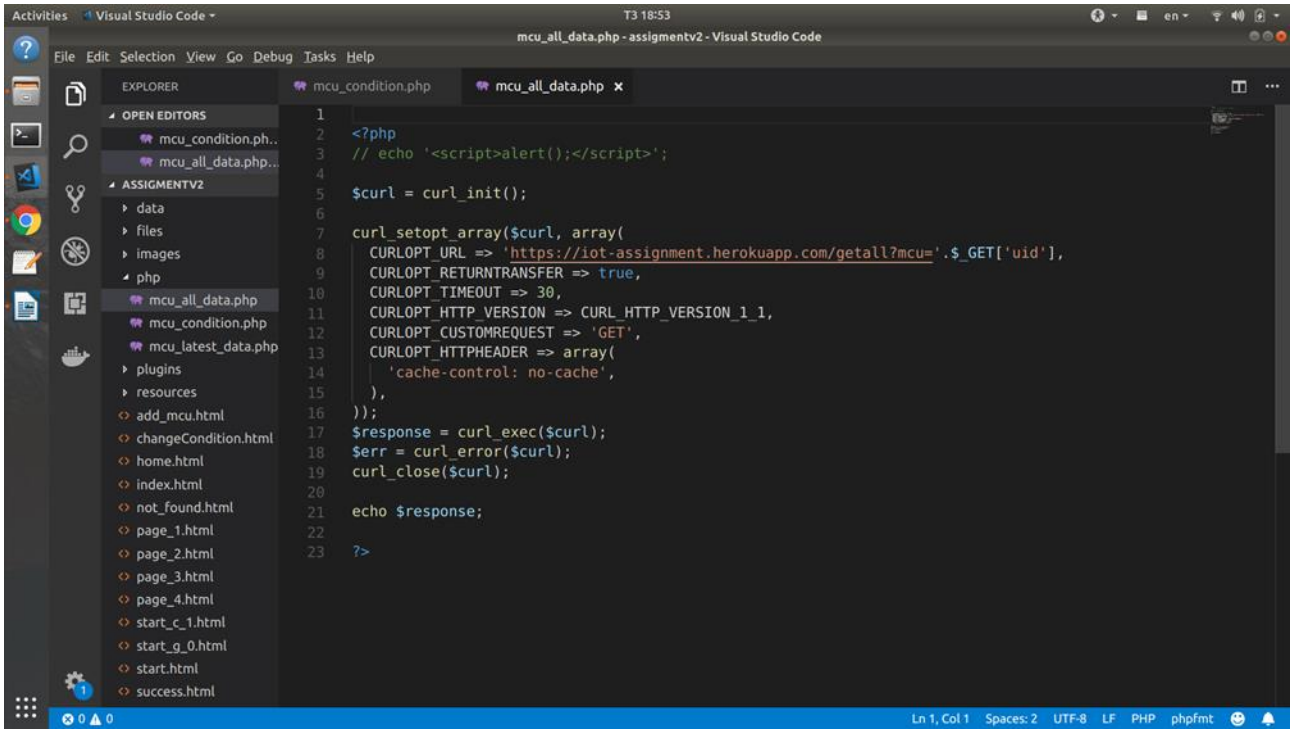
- Get MCU condition



- Get latest data of MCU

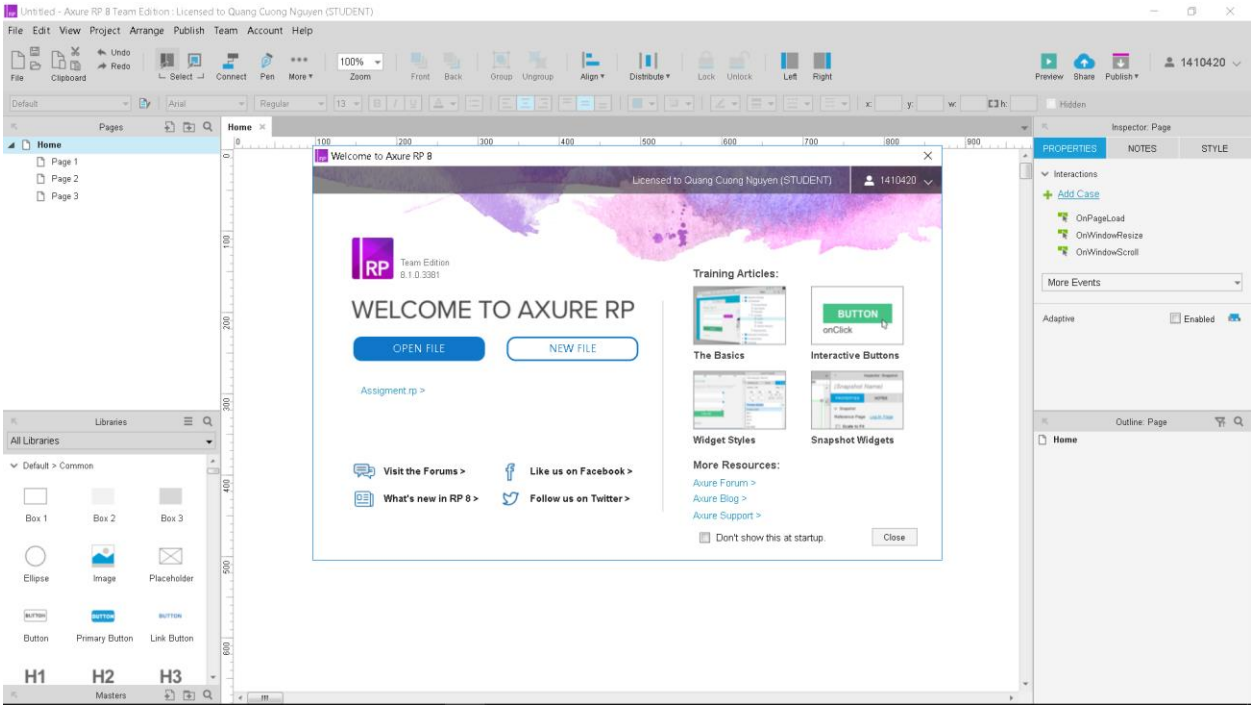


- Get all data of MCU for view logs

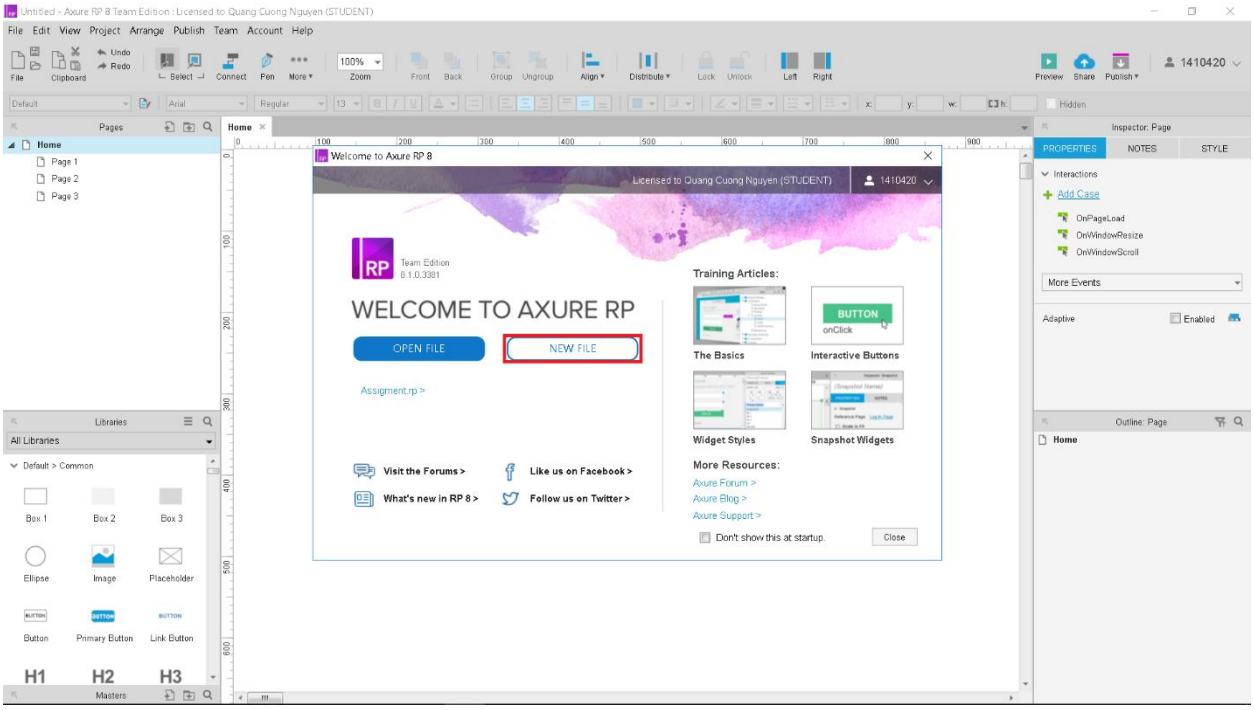


3. UI

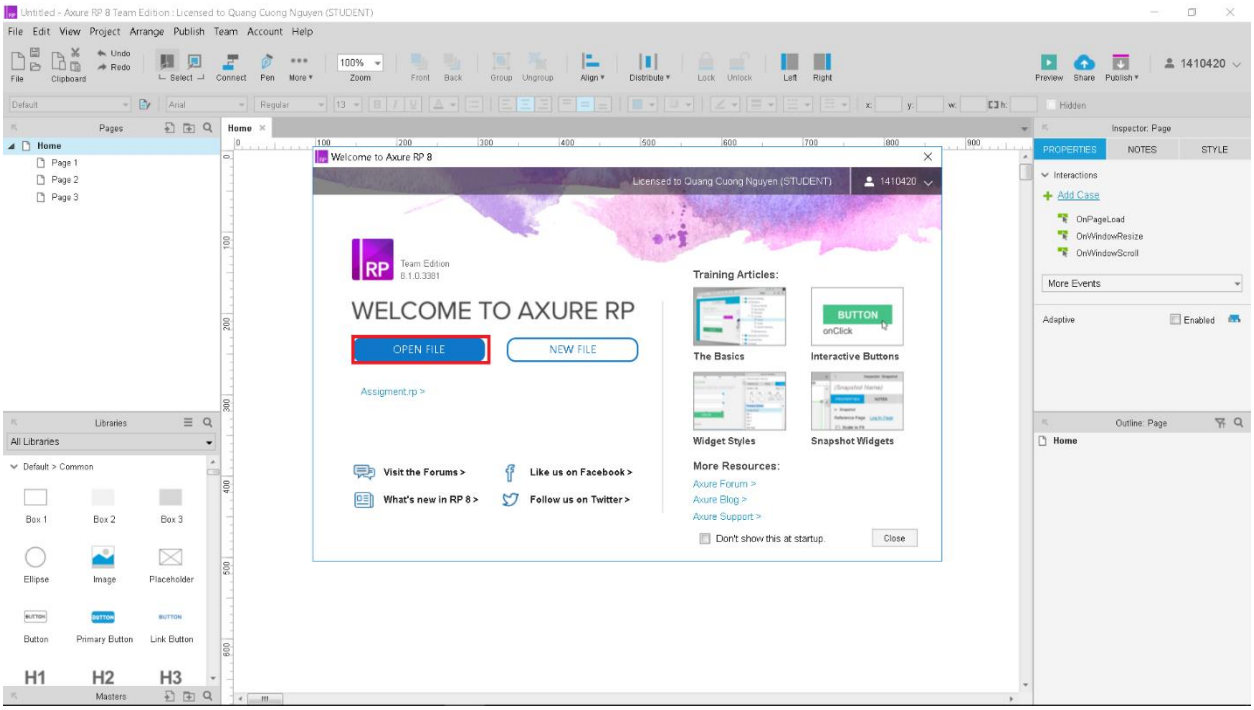
- Download Axure RP8 at https://www.axure.com/download
- Open Axure RP



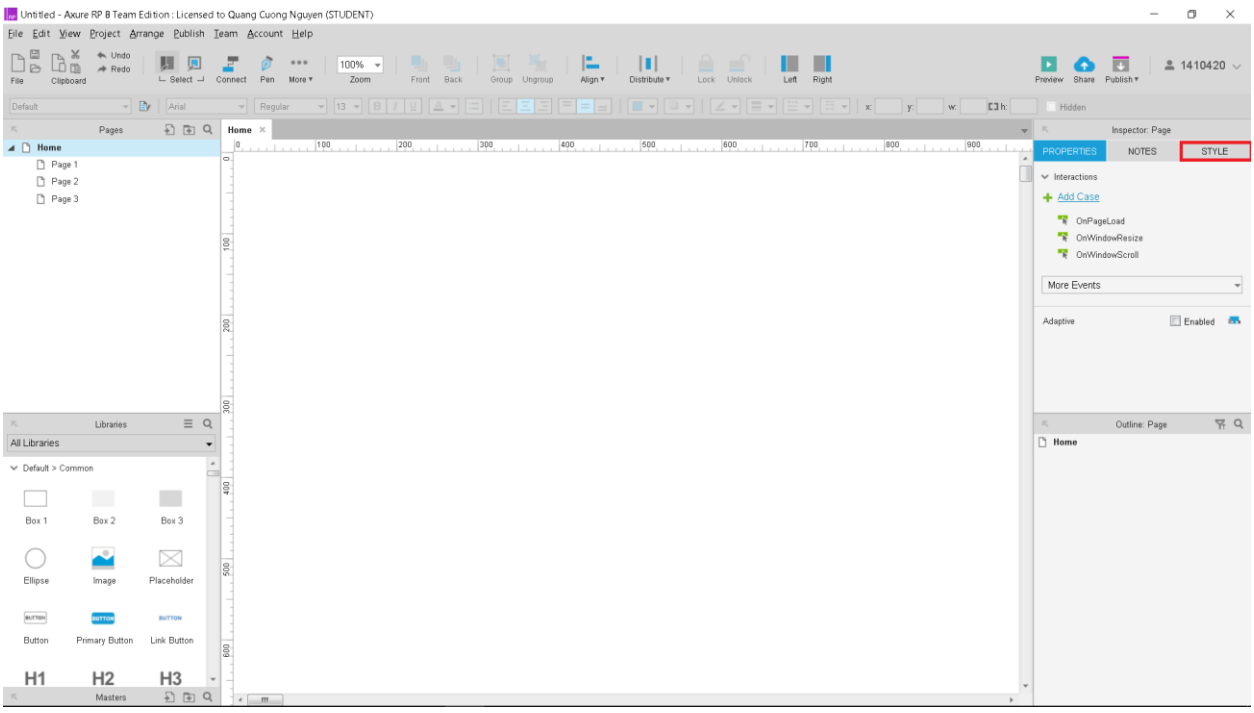
- Create new file



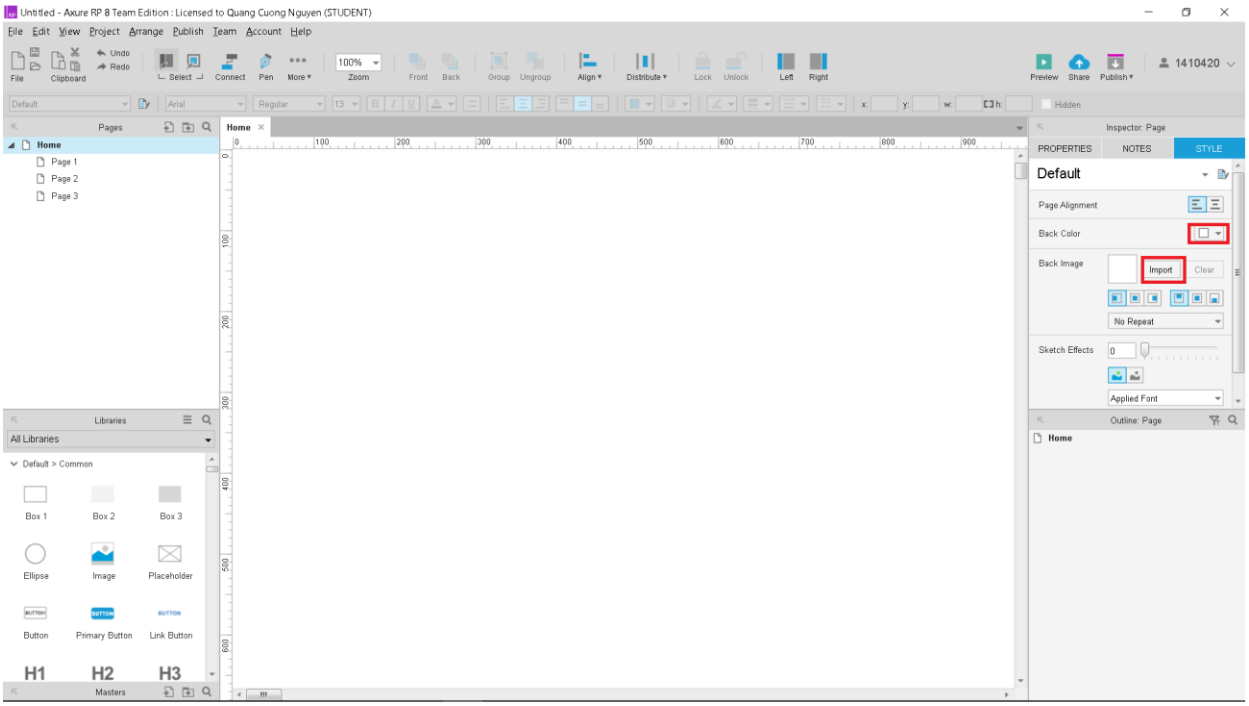
- Create new file



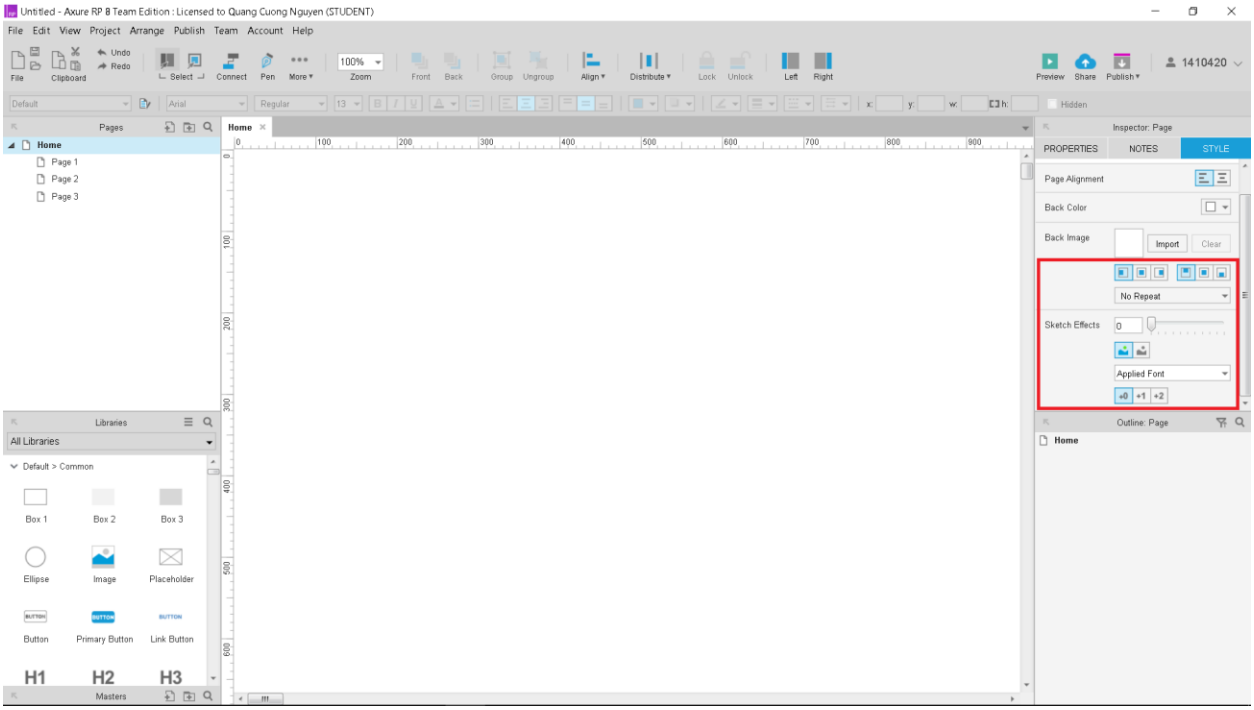
- Open existed file
- Design background



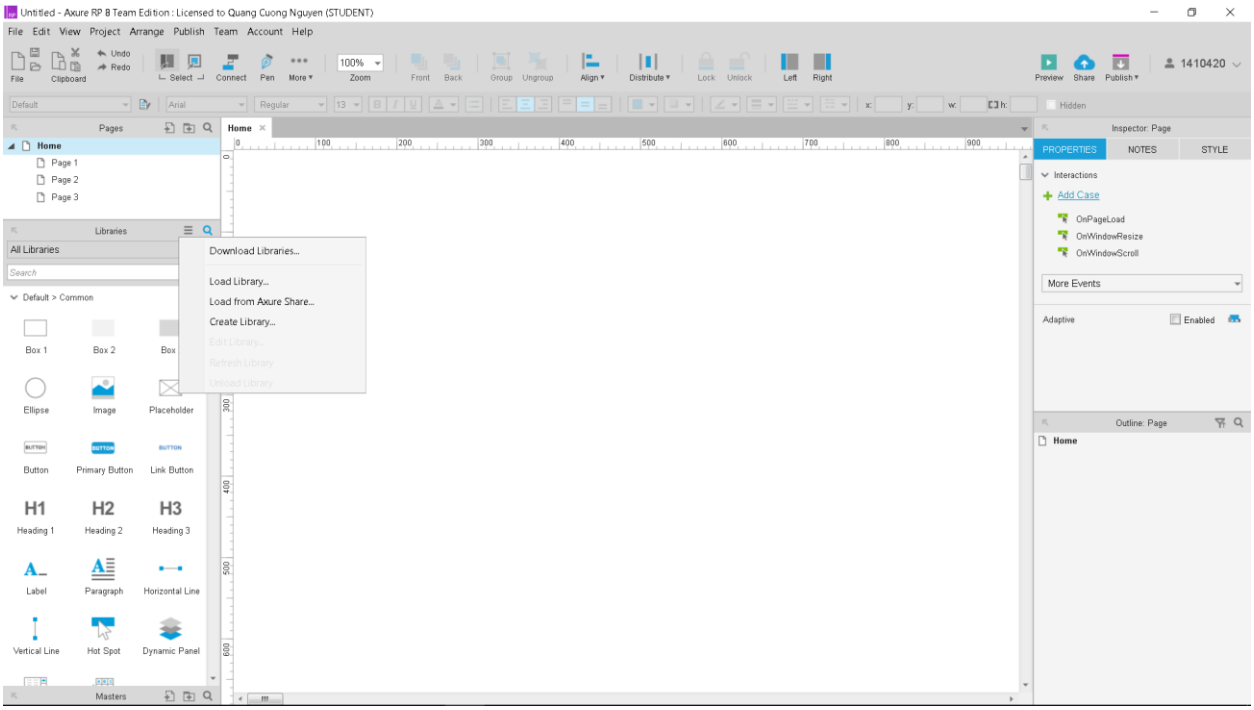
- Click Style



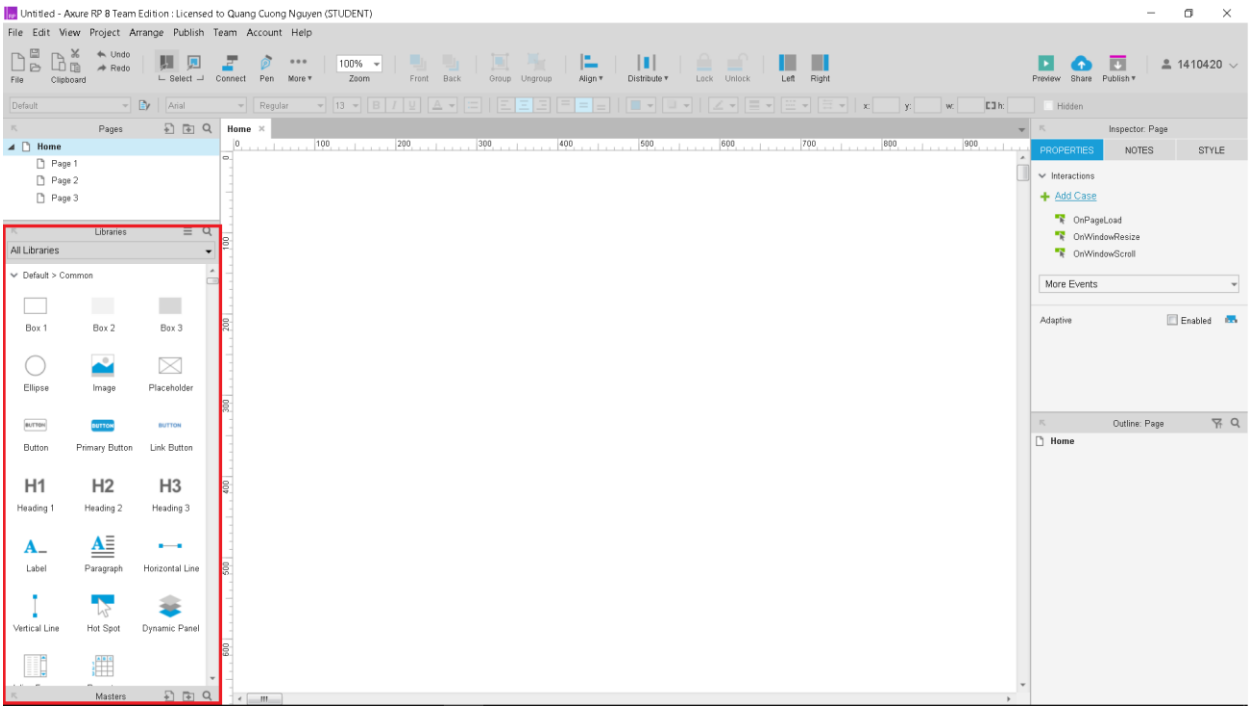
- Choose color or Import picture for background



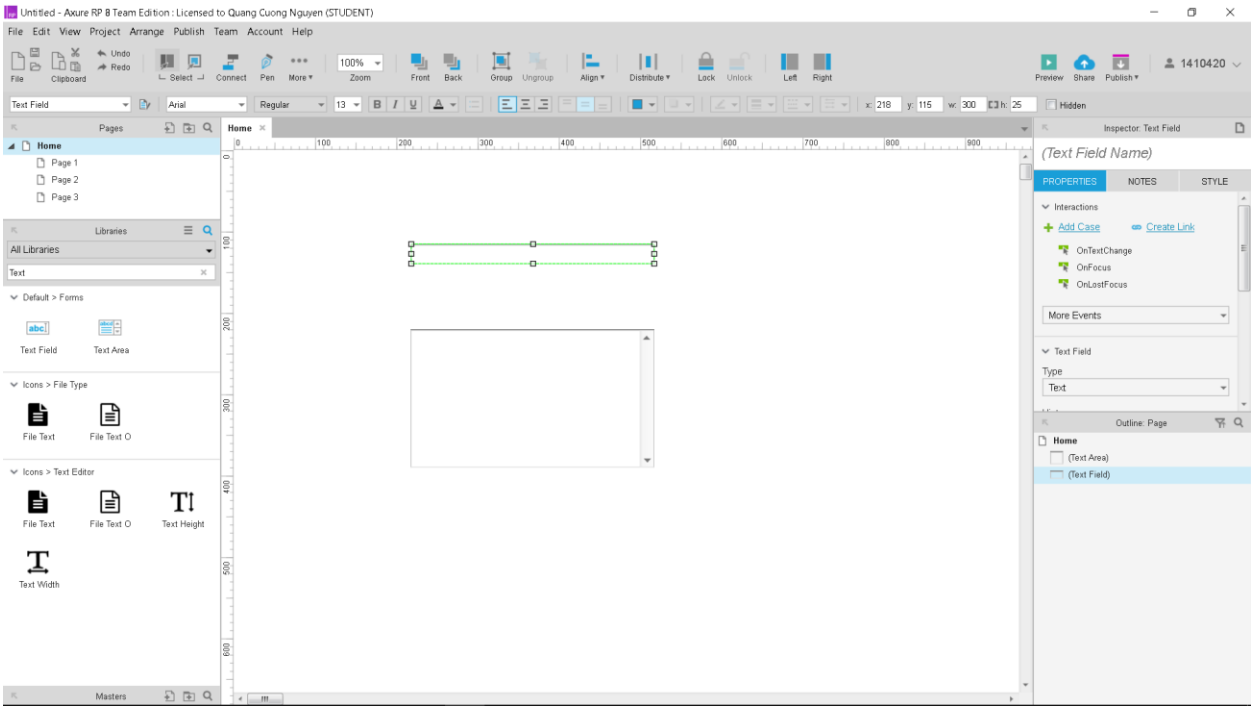
- Edit background
- Design text field / area



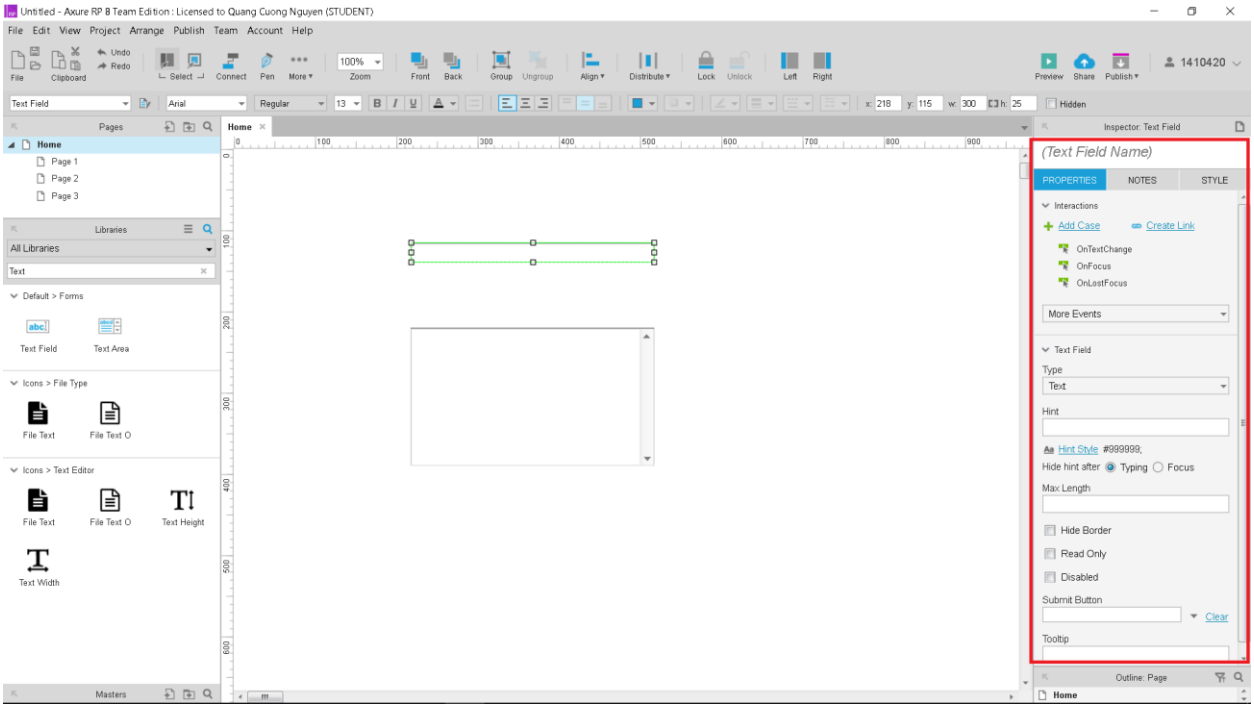
- Download and add more libraries



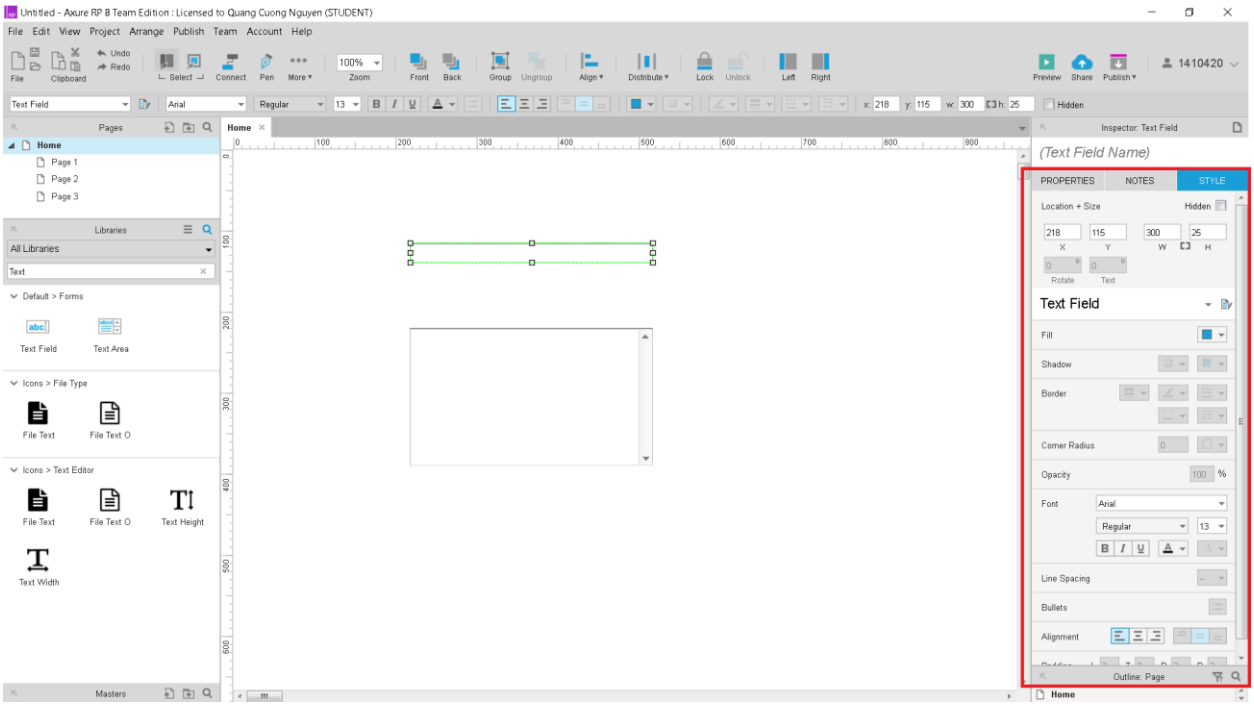
- Object for design website



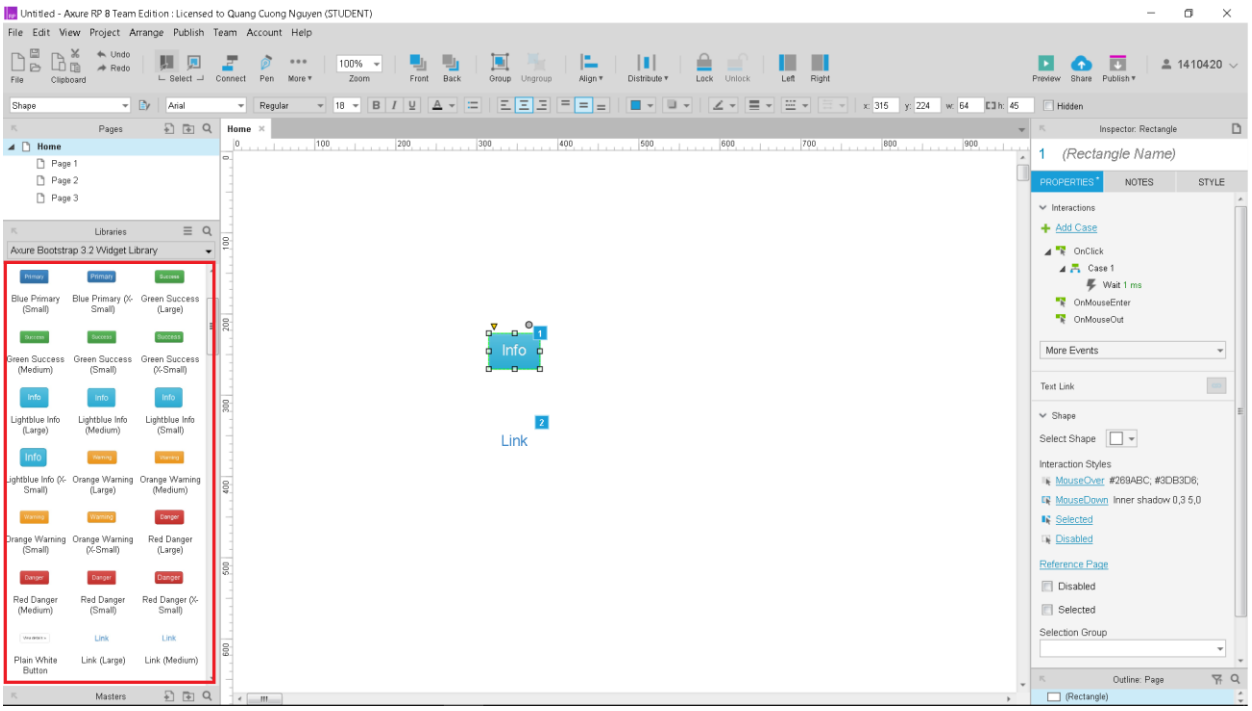
- Create text field and text area



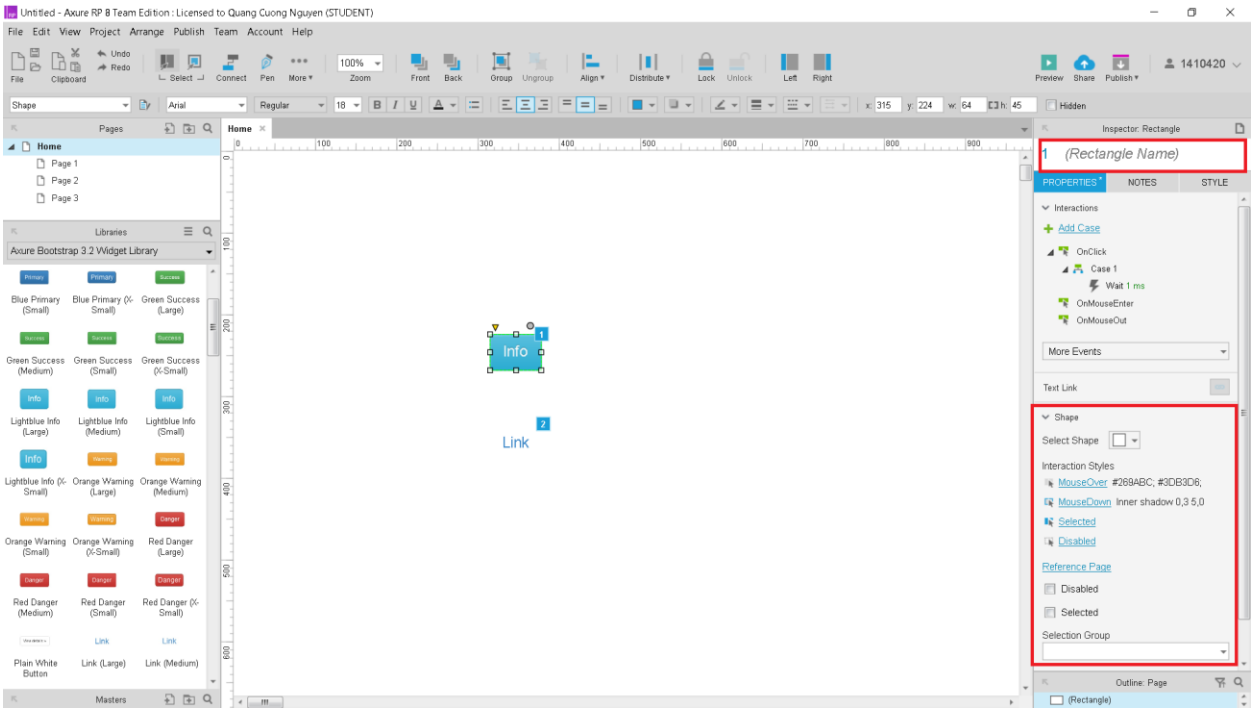
- Edit name and features



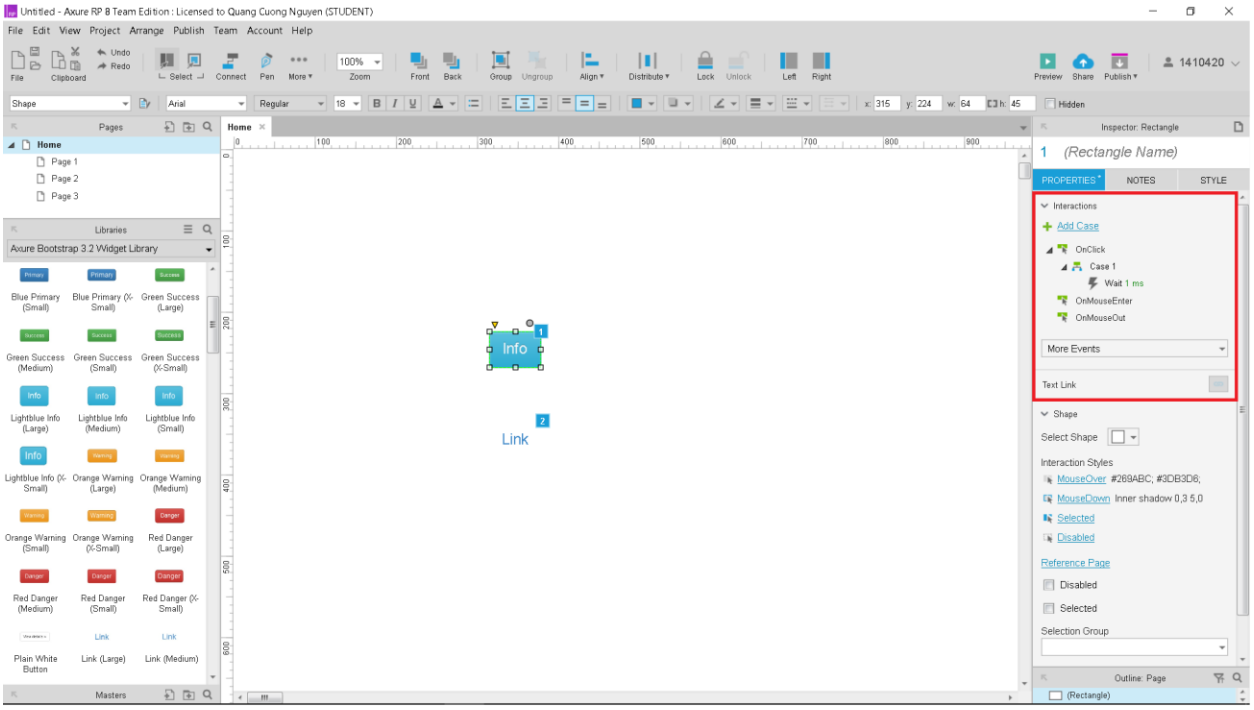
- Edit style (font, format…)
- Design button and link



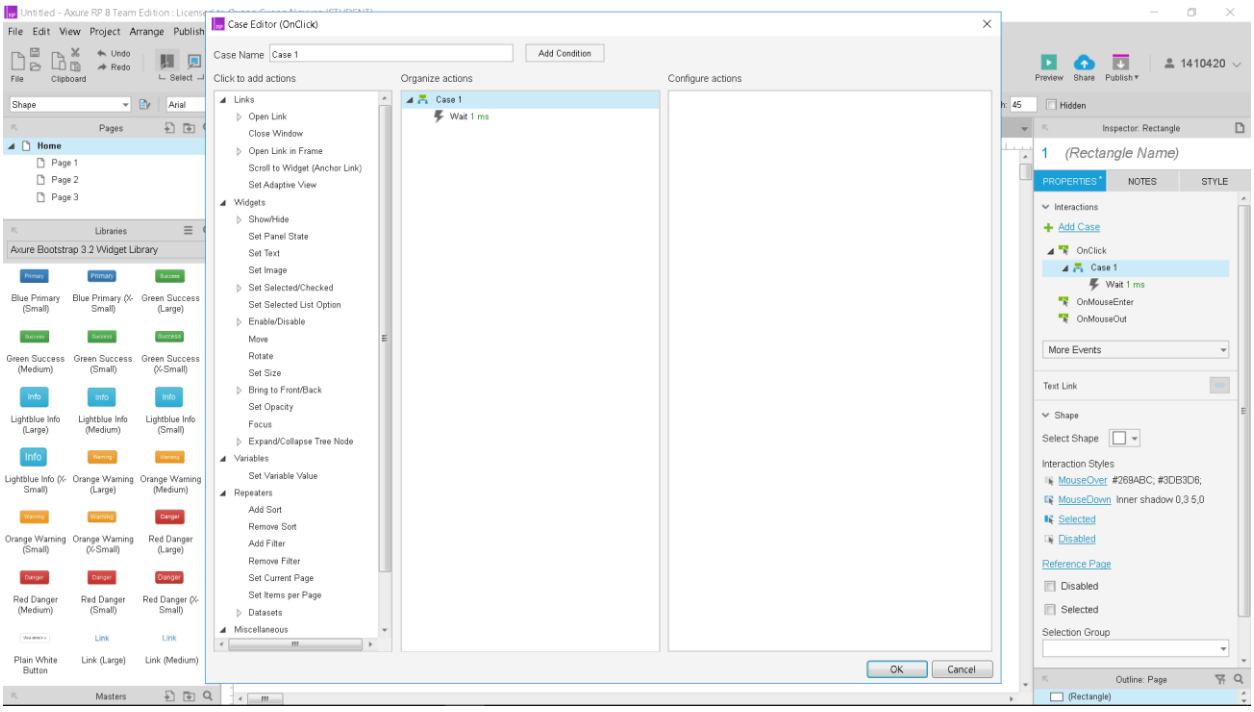
- Some buttons and links



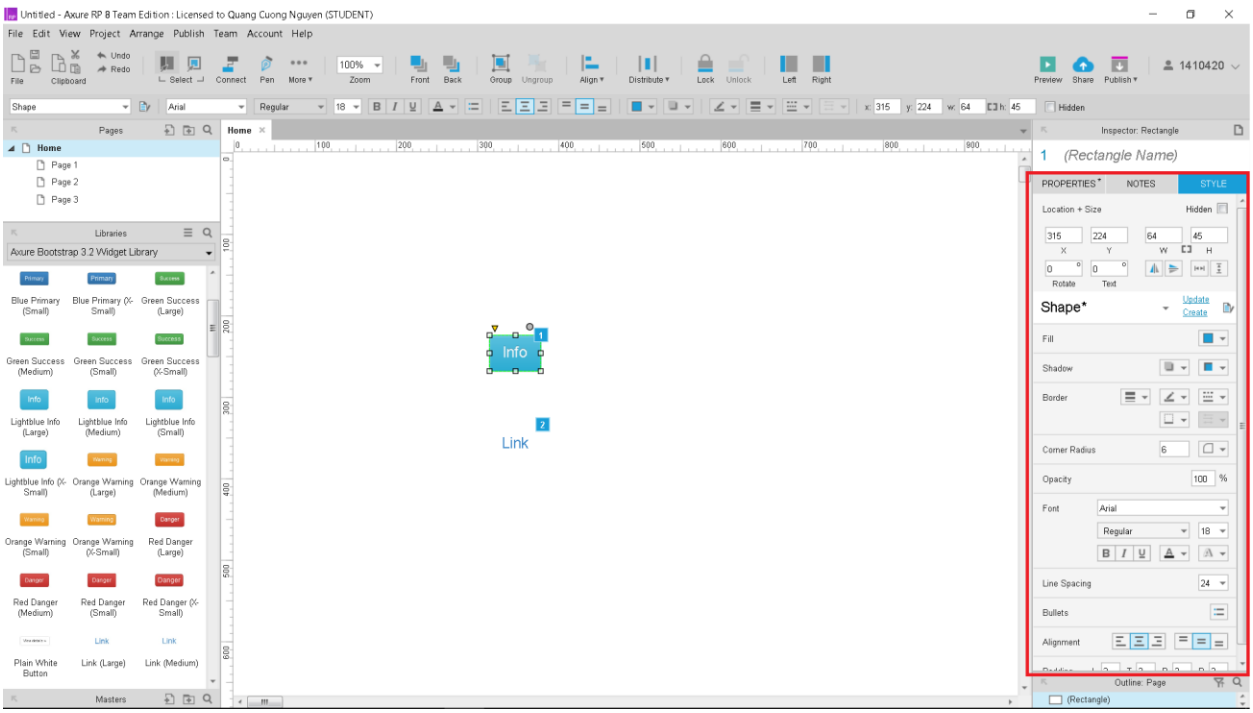
- Edit name and features



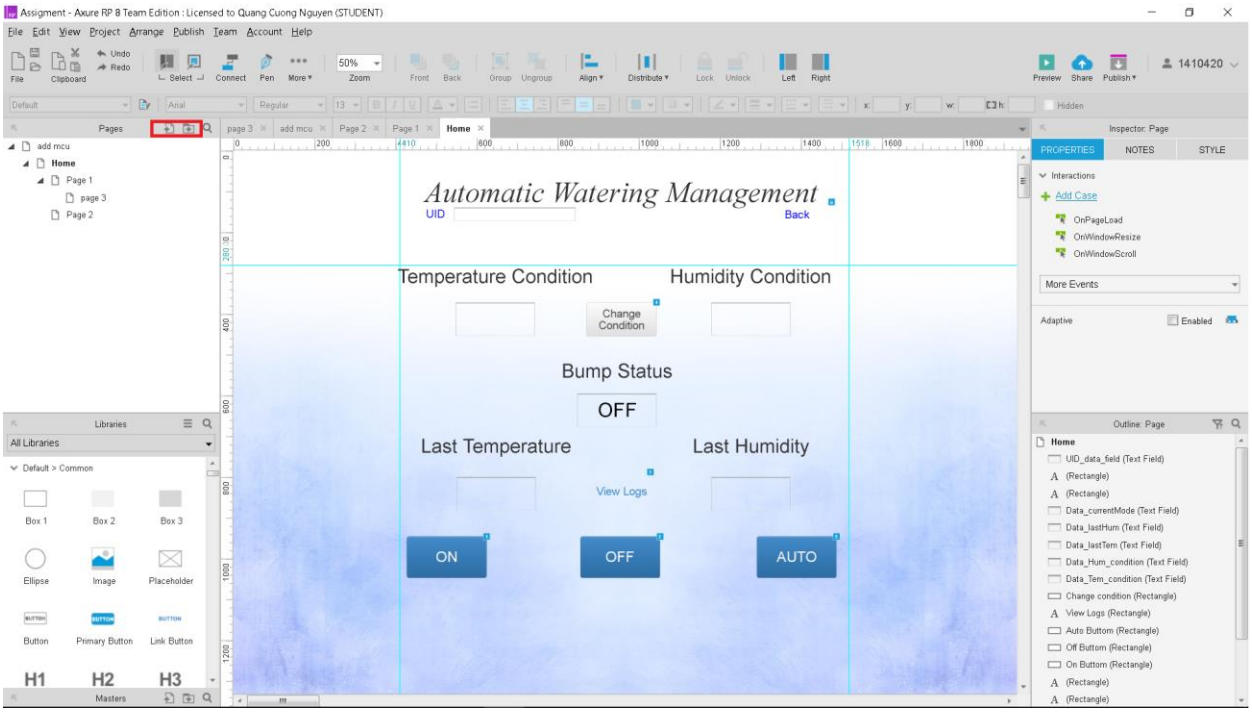
- Edit actions



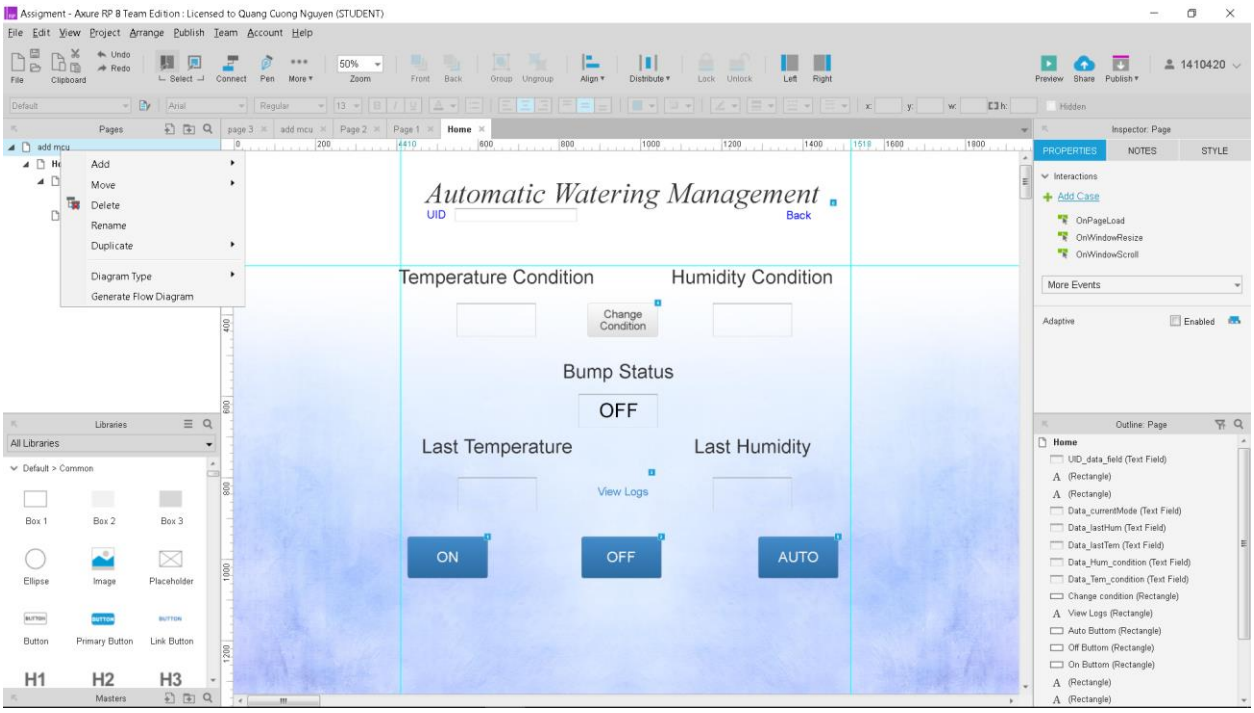
- Example: some action when you click on the button/link



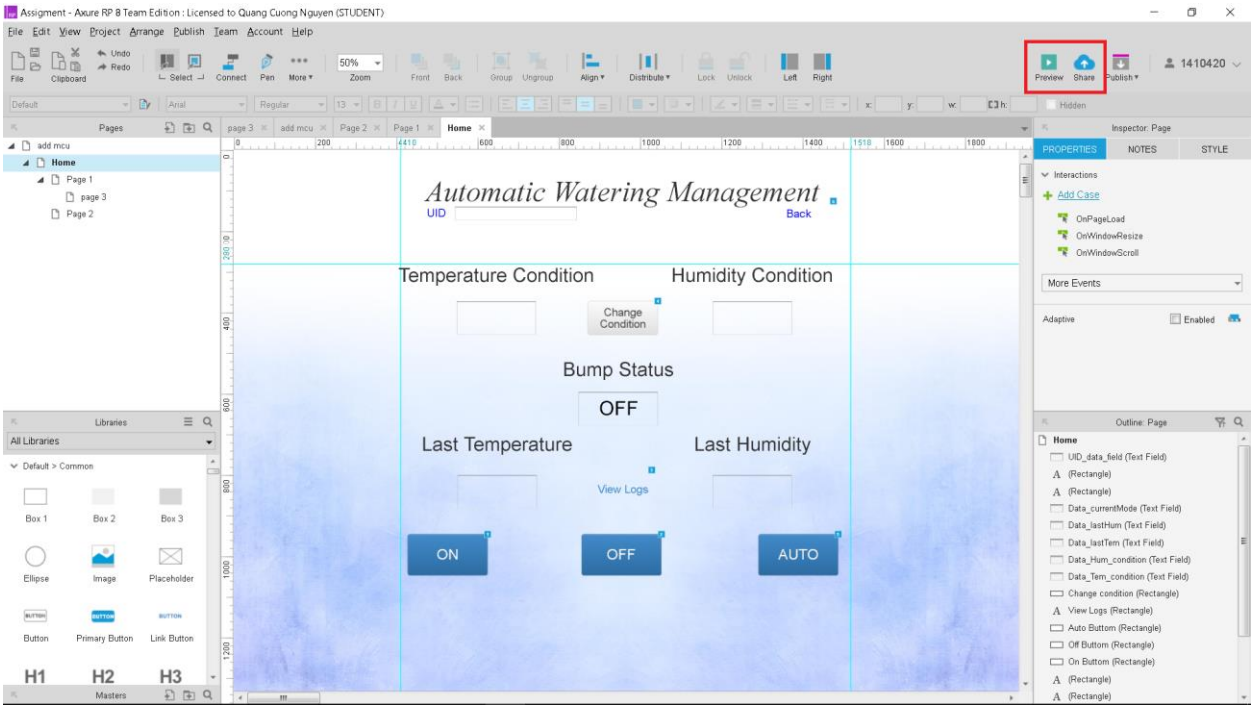
- Edit style
- Edit page



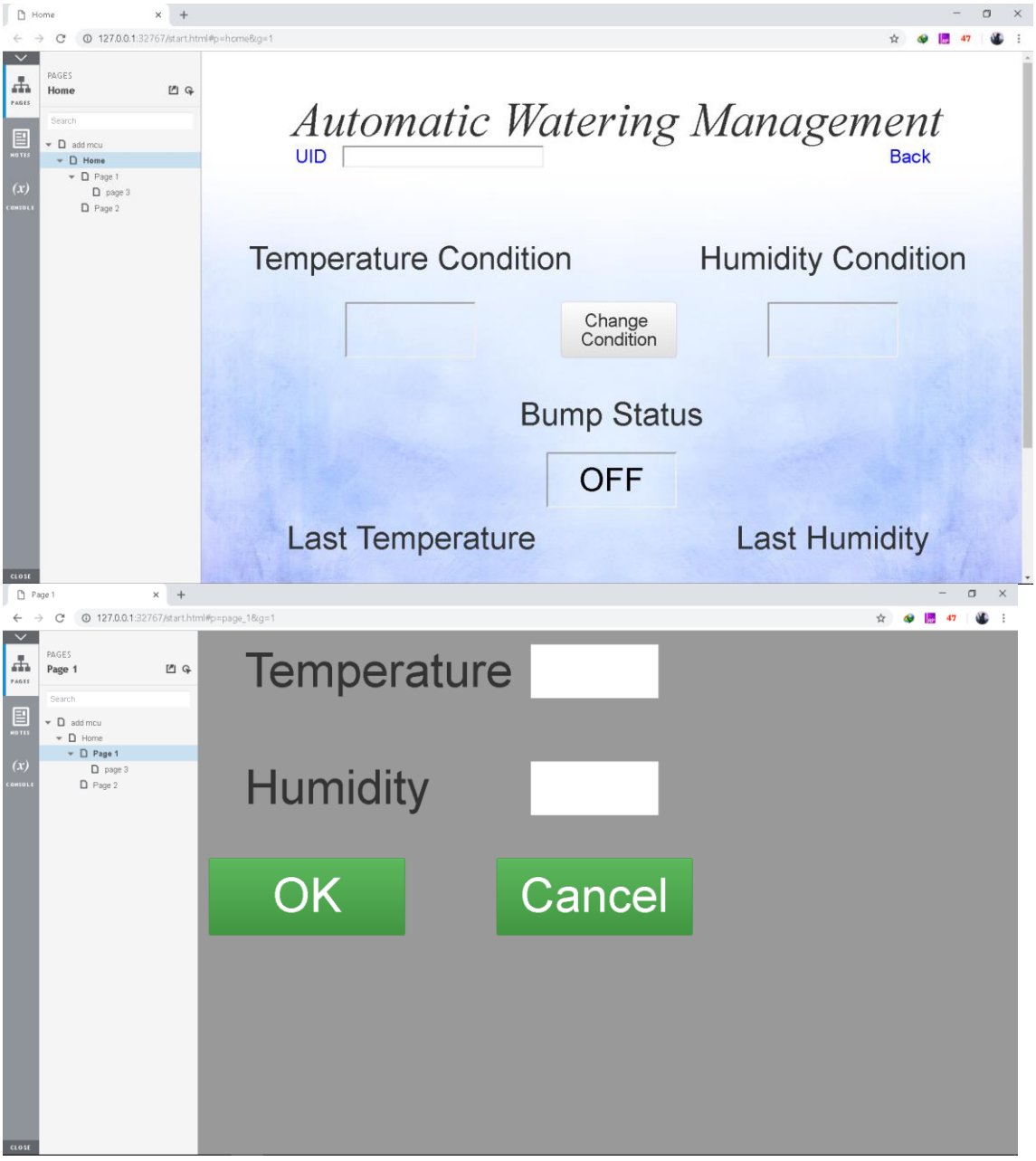
- Add page/folder

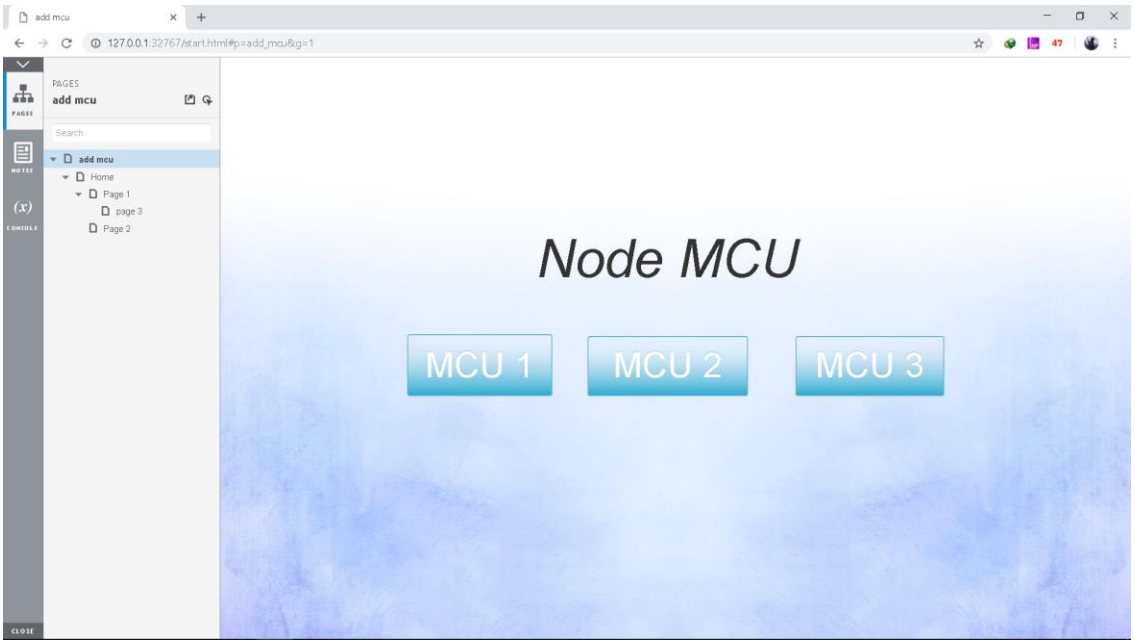


- Right click to edit page
- Preview and share



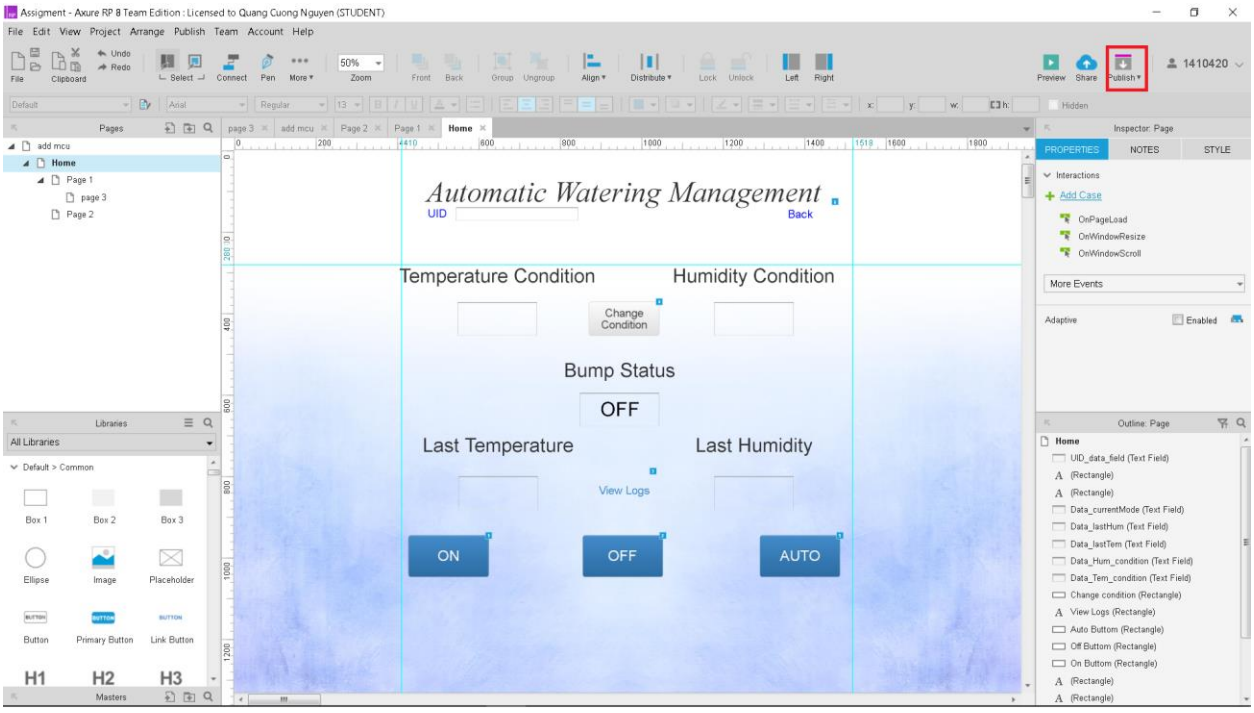
- Click to preview or share


Home
127.0.0.1:32767/start.html#p=home&g=1
PAGES
Home
Search
add mcu
Home
Page 1
page 3
Page 2
Automatic Watering Management
UID
Back
Temperature Condition
Humidity Condition
Change Condition
Bump Status
OFF
Last Temperature
Last Humidity
Page 1
127.0.0.1:32767/start.html#p=page_1&g=1
PAGES
Page 1
Search
add mcu
Home
Page 1
page 3
Page 2
Temperature
Humidity
OK
Cancel

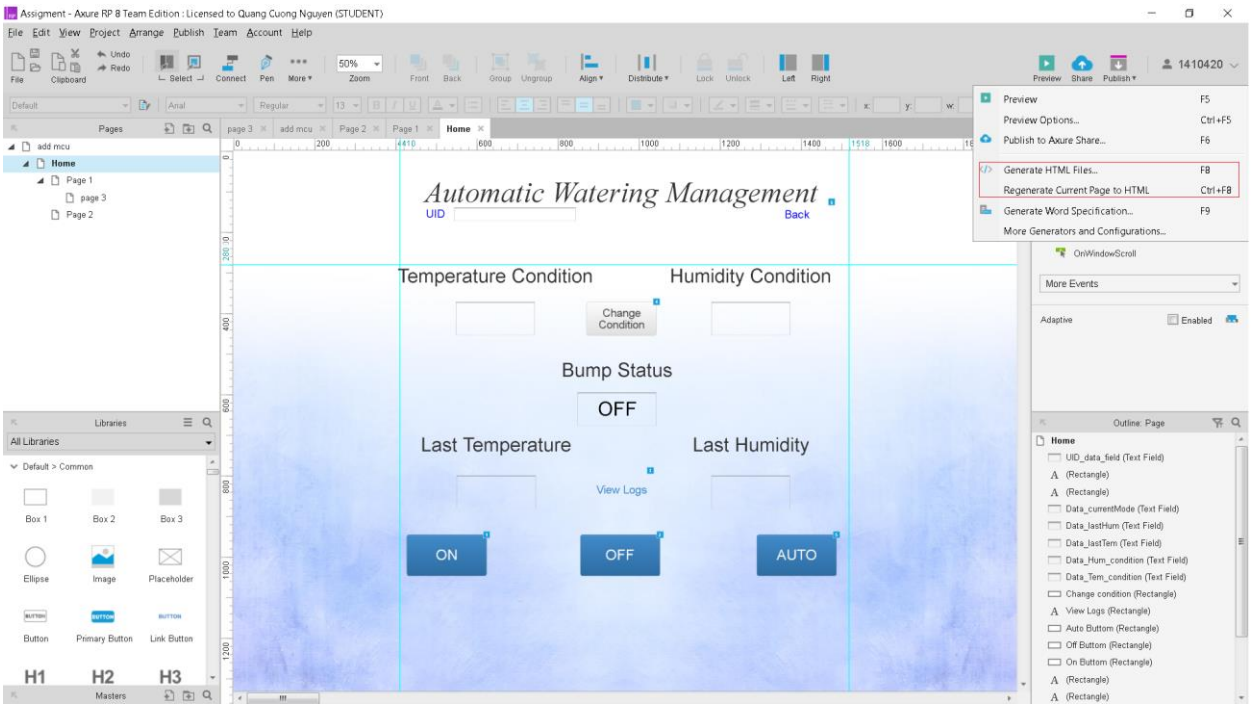


Example of preview

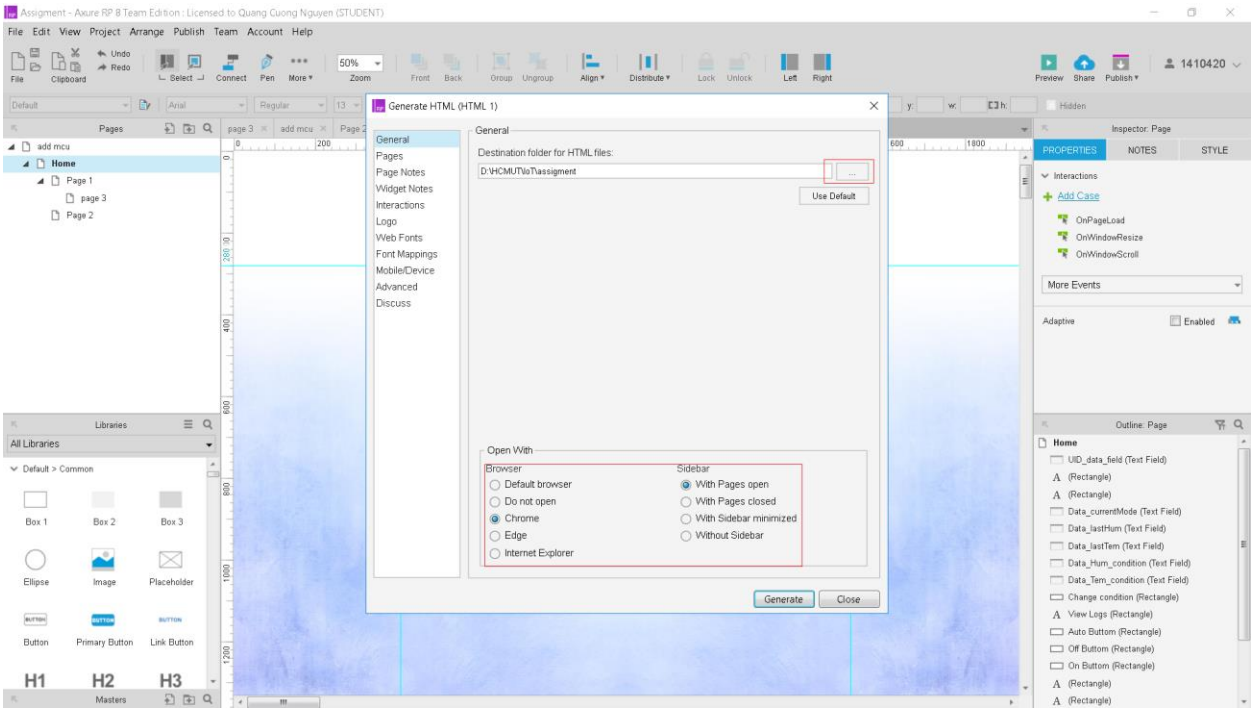
- Convert to html code



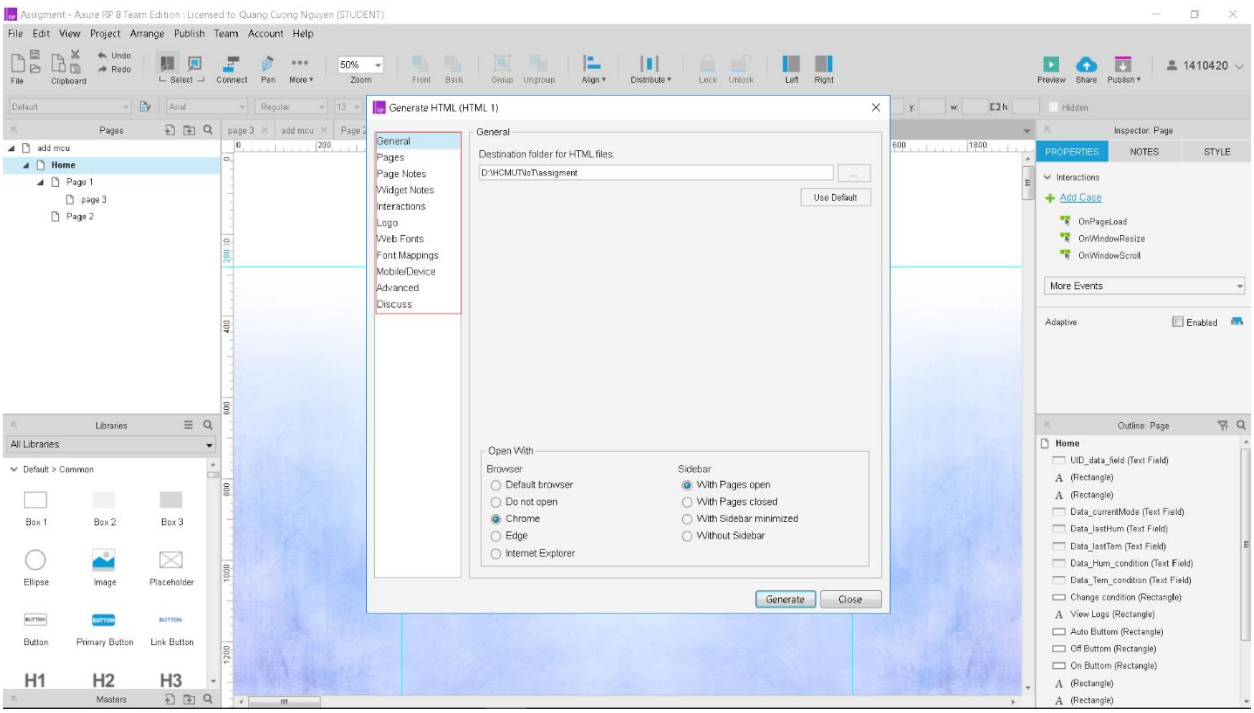
- Left clink on Publish



- F8 for convert all pages, Ctrl+F8 for convert current page



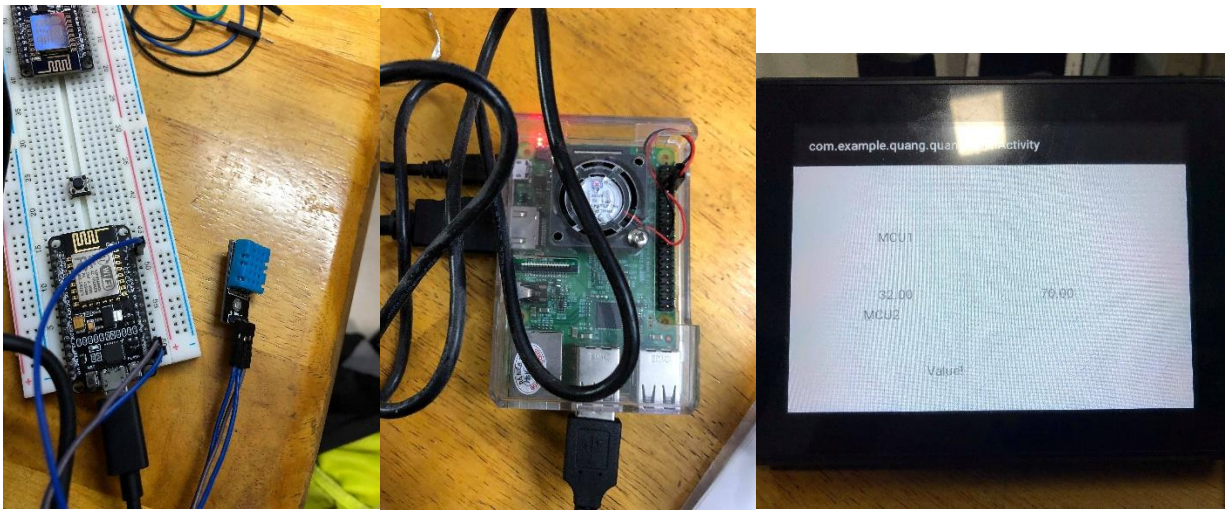
- Click on "…" to choose place to save, edit some features at "open with" box



- Edit more features by click on left column and click on Generate to finish

IV. DEMO

1. Hardware

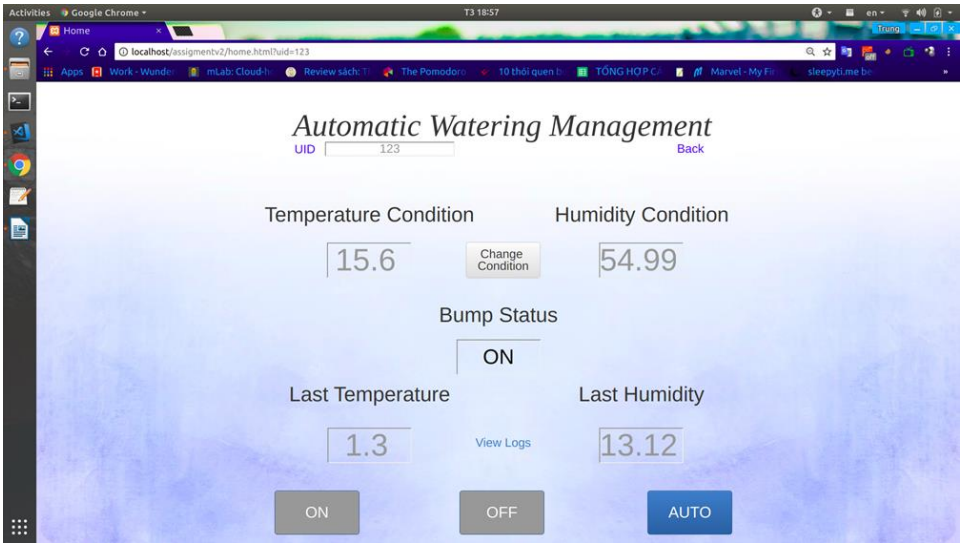


DHT-11 connects to node MCU esp8266.
Raspberry Pi runs Android Things.

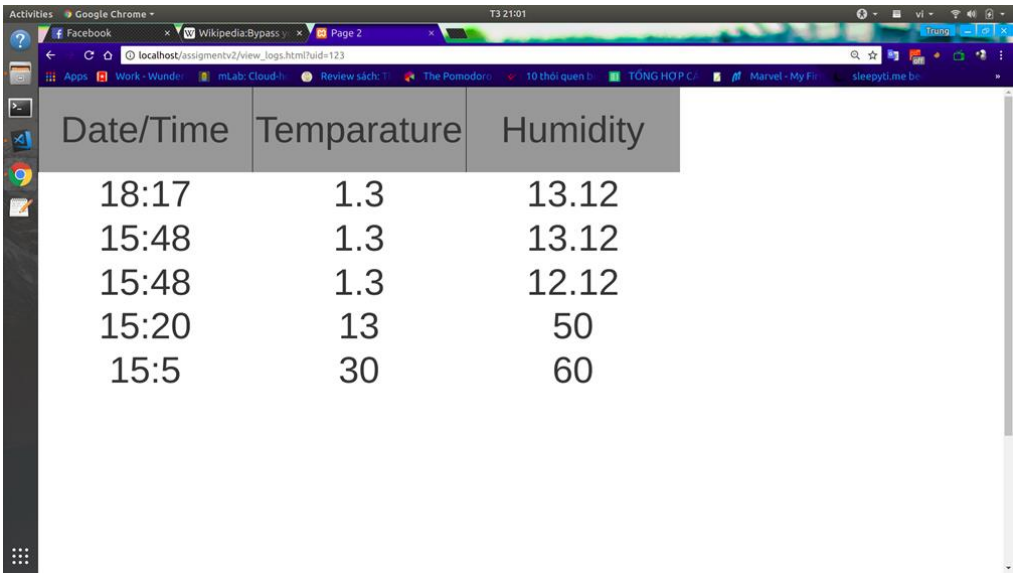
Small UI for checking the data.

2. Server

In home screen with example mcu:

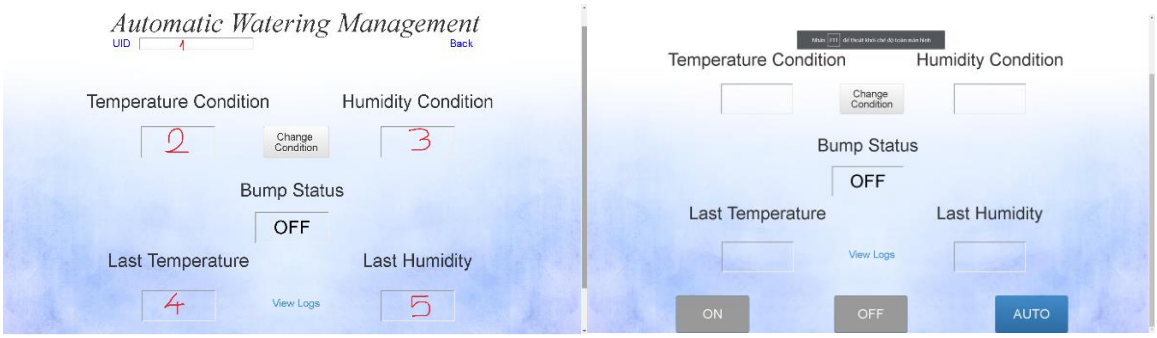


View logs example



And all buttons and functions in pages are working correctly.

3. Website UI



- UID (1) is ID of the MCU.
- Temperature condition (2) is number that the bump will automatically spray if temperature of land risk to.
- Humidity condition (3) is number that bump will automatically spray if humidity of land fall to.
- Last temperature (4) is temperature of land at the last time when DHT-11 got.
- Last humidity (5) is humidity of land at the last time when DHT-11 got.
- Bump status is the status of the bump (On/Off) at current.
- Change condition button, use for edit temperature condition and humidity condition. With this button, the user can edit temperature and humidity.
- View logs link, use to open the view logs page where the monthly data save at.
- On/Off/Auto button, these button use for change bump status, on/off button use to open/close the bump immediately. Auto button will control the bump base on humidity condition and temperature condition.