Summer Fellowship Report

On

UI Improvements, Bug fixes and Angular Version Upgrade

Submitted by

Asif Sajid Mulla

Dr. D.Y. Patil Pratishthan's College of Engineering,Salokhenagar Kolhapur.

Under the guidance of

Prof. Kannan Moudgalya Chemical Engineering Department IIT Bombay

Mentors

Mr. Nagesh Karmali

Ms. Firuza Aibara

Acknowledgement

I, the summer intern of the FOSSEE - Arduino On Cloud, am so overwhelmed in all humbleness and gratefulness to acknowledge my deep gratitude to all those who have helped me put my ideas to perfection and have assigned tasks well above the level of simplicity and into something concrete and unique.

I wholeheartedly thank **Prof. Kannan M. Moudgalya** for having faith in me, selecting me to be a part of his valuable project and for constantly motivating me to do better. I thank **Mr. Nagesh Karmali** and **Ms. Firuza Aibara** for providing me the opportunity to work on this project. I am also very thankful to my mentors for their valuable suggestions. They were and are always there to show me the right track when needed help. With help of their brilliant guidance and encouragement, I was able to complete tasks properly and was up to the mark in all the tasks assigned. During the process, i got a chance to see the stronger side of our technical and nontechnical aspects and also strengthen our concepts. Last but not the least, I sincerely thank all our other colleagues working in different projects under **Prof. Kannan M. Moudgalya** for helping us evolve better with their critical advice.

Declaration

I declare that this written submission represents our ideas in my own words and whenever other's ideas or words have been included, I have adequately cited and referenced the original sources. I declare that I have properly and accurately acknowledged all sources used in the production of this thesis.

I also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources which have not been properly cited or from whom proper permission has not been taken when needed.

Asif Mulla

Index

List Of Figures

Chapter 1

Introduction

Arduino on cloud is a cloud-based simulator, which can be used by students and researchers to test and run the simulation on the web, before actual execution. This system allows the users to drag and drop Arduino components from the left pane onto the working space on the right. The pins of the Arduino board can be connected to various input/ output devices like LED, motor, push-button, etc using wires. There is also a facility to change the color of wires, LEDs, and such components, to differentiate easily. The users can then proceed to write their code in the code window which is then simulated. There is an option for the users to print or save it in pdf format for documentation purposes. The basic ERC check enables the users to find out errors if any.[1]

1.1. Project Overview

During this fellowship, I worked on the existing project which was already completed to some extent, for the most part, I did login bug fixes, updating angular version, and circuit additions in the gallery and features added. Improvements some Features.

Chapter 2

Improvement UI

During the internship, I added a few features to the project, most of these features aimed at improving the accessibility of the website and facilitate some stuff in the website.

2.1 Dashboard UI.

The previous dashboard contained only tabs mode which included two tabs namely on cloud and temporary circuits.I changed the UI by adding a sidebar which contains the user profile details and their respective roles.this sidebar is reusable. The home page is also dynamic. Also,the navbar consisted of only the image in the rightmost corner, which now includes the initial of the users.

One more feature added is the temporary UI had a separate tab previously which is now converted into a separate page.

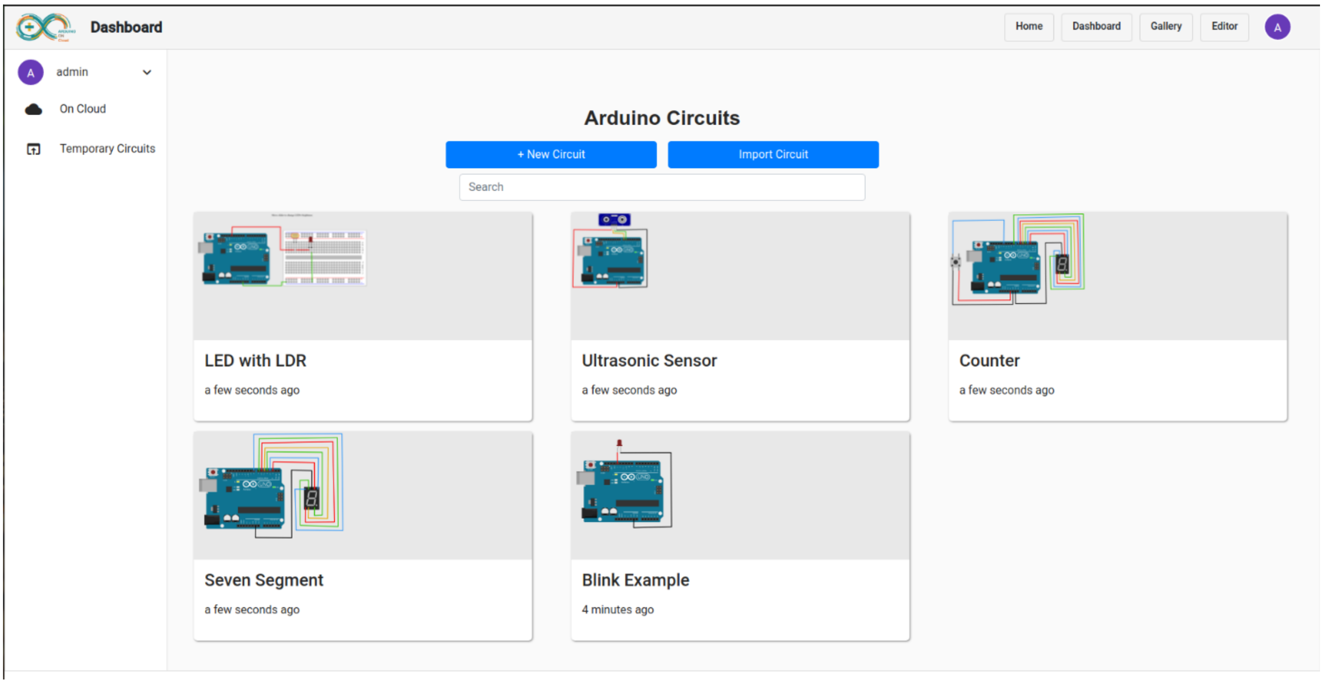
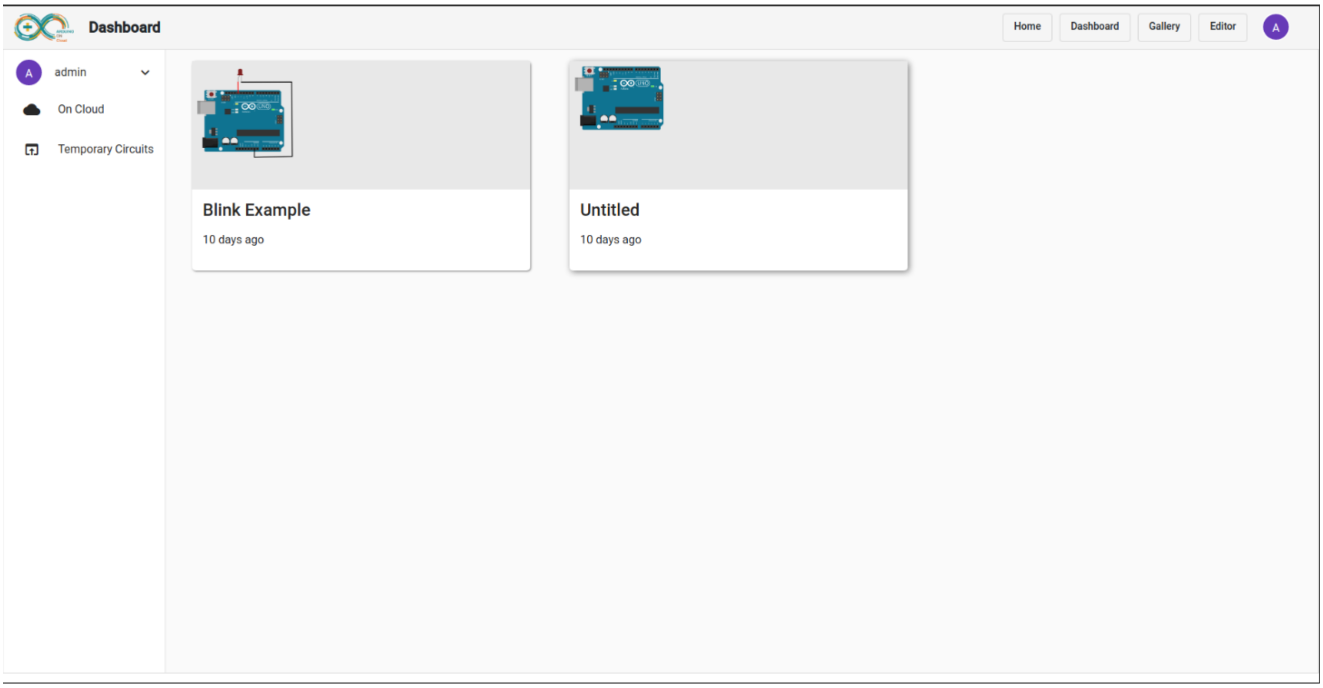


Figure 2.1: Dashboard UI.



Blink Example
10 days ago

Untitled
10 days ago

Figure 2.1.1: Temporary Circuit UI.

2.2 Gallery UI

The previous UI opened the gallery in the same window but with a different UI. I modified this UI by adding the gallery in the same Window and same UI.

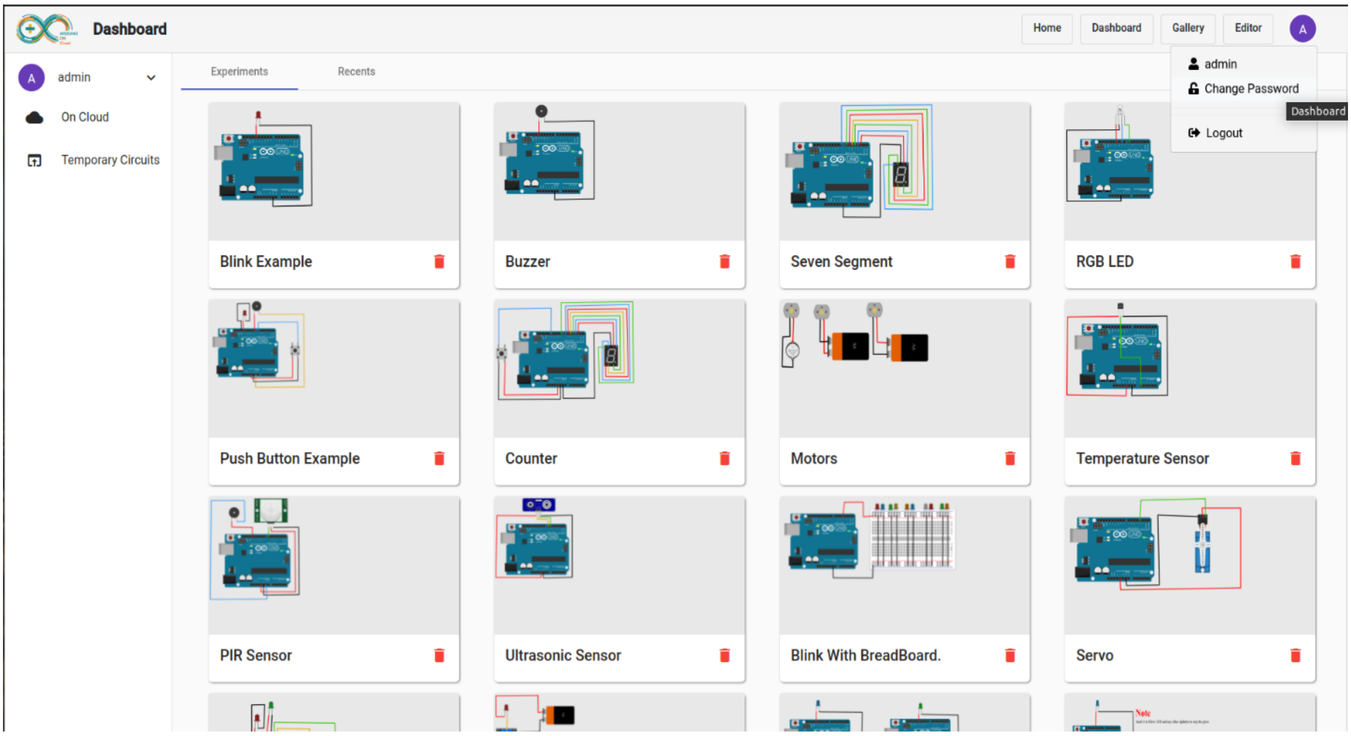
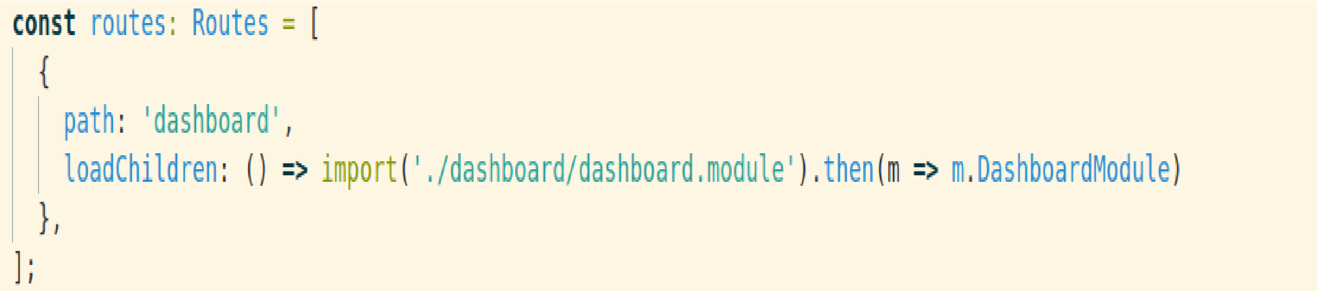


Figure 2.2: Gallery UI.

2.3 Lazy Loading Features.

Lazy loading is the process of loading components, modules, or other assets of a website as they're required.Since Angular creates a SPA (Single Page Application), all of its components are loaded at once. This means that a lot of unnecessary libraries or modules might be loaded as well.For a small application this would be okay. But as the application grows the load time will increase if everything is loaded at once. Lazy loading allows Angular to load components and modules as and when they're needed.



```
const routes: Routes = [
  {
    path: 'dashboard',
    loadChildren: () => import('./dashboard/dashboard.module').then(m => m.DashboardModule)
  },
];
```

2.3 Lazy Loading Features

Chapter 3

Update Angular Version

The previous version of the arduino cloud angular version was 7.0. This was a new challenge for me as I had not worked for such before; so it seemed more interesting to work on. I updated the version one by one (i.e. 7,8,9,10,11).

Passing through each of the phases of the version. Angular syntax was changed.which i checked whether it was correct or not after successful migration of each version. I runned the development as well as production mode.

3.1 Update angular 7 to 11 LTS.

 I overloaded the project structure before actual work of modification, where I found the docker node version was 10. This seemed incompatible while updating the version. So i updated the docker node to 14.0

When I completed the updates till version 11, I found the two packages (isundefined , is null) were deprecated from angular 4. Thus, I modified the code to make it work smoothly in future.

Steps For Updating angular 7 to 11 LTS.

1.Before Updating

If you use the legacy HttpModule and the Http service, switch to HttpClientModule and the HttpClient service. HttpClient simplifies the default ergonomics (you don't need to map to JSON anymore) and now supports typed return values and interceptors. Read more on angular.io.

2.During the Update

1. Update to version 8 of the core framework and CLI by running ng update @angular/cli@8 @angular/core@8 in your terminal and review and commit the changes.

2. Replace /deep/ with ::ng-deep in your styles, read more about angular component styles and ::ng-deep. /deep/ and ::ng-deep both are deprecated but using ::ng-deep is preferred until the shadow-piercing descendant combinator is removed from browsers and tools completely.

3. Angular now uses TypeScript 3.4, read more about errors that might arise from improved type checking.

4. The CLI's build command now automatically creates a modern ES2015 build with minimal polyfills and a compatible ES5 build for older browsers, and loads the appropriate file based on the browser. You may opt-out of this change by setting your target back to es5 in your tsconfig.json.

5. When using new versions of the CLI, you will be asked if you want to opt-in to share your CLI usage data. You can also add your own Google Analytics account. This lets us make better decisions about which CLI features to prioritize, and measure the impact of our improvements.

6. If you use ViewChild or ContentChild, we're updating the way we resolve these queries to give developers more control. You must now specify that change detection should run before results are set. Example: @ContentChild('foo', {static: false}) foo !: ElementRef;. ng update will update your queries automatically, but it will err on the side of making your queries static for compatibility.

7. Update Angular Material to version 8 by running ng update @angular/material@8 in your terminal.Instead of importing from @angular/material, you should import deeply from the specific component. E.g. @angular/material/button. ng update will do this automatically for you.

8. For lazy loaded modules via the router, make sure you are using dynamic imports. Importing via string is removed in v9. ng update should take care of this automatically.

9. Make sure you are using Node 10.13 or later.

10 .Run ng update @angular/core@9 @angular/cli@9 which should bring you to version 9 of Angular.

11. Your project has now been updated to TypeScript 3.8, read more about new compiler checks and errors that might require you to fix issues in your code.

12. Run ng update @angular/material@9

13. If your project depends on other Angular libraries, we recommend that you consider updating to their latest version. In some cases this update might be required in order to resolve API incompatibilities. Consult ng update or npm outdated to learn about your outdated libraries.

14. Angular 9 introduced a global $localize() function that needs to be loaded if you depend on Angular's internationalization (i18n). Run ng add @angular/localize to add the necessary packages and code modifications. Consult the $localize Global Import Migration guide to learn more about the changes.

15. Make sure you are using Node 12 or later.

16. Run ng update @angular/core@10 @angular/cli@10 which should bring you to version 10 of Angular.

17. Run ng update @angular/material@10

18. New projects use the filename .browserslistrc instead of browserslist. ng update will migrate you automatically.

19. Run ng update @angular/core@11 @angular/cli@11 which should bring you to version 11 of Angular.

20. Run ng update @angular/material@11.

21.Support for IE9, IE10, and IE mobile has been removed. This was announced in the v10 update.

Chapter 4

Arduino staff Role

In e-Sim,a staff role was already present which gave authority to staff for saving the circuit in the gallery. This feature was not available in arduino so, I added this feature in arduino by name Arduino staff role. The API's for staff role were already added.

I consumed them to construct the same. I added the staff arduino role group key. We can check whether the platform is arduino or e-sim to save the circuits.

4.1. Save circuit on gallery

When a user logins in an arduino project, we receive a token after successful login after receiving the token. We passed it to getRoles API where we send the roles accordingly to the user. If the user has a staff role assigned then the add to gallery button is enabled and he/she can save the image by clicking on it.
I used a data dump image and circuit name for the API, using which I can write http verbs on the client side.

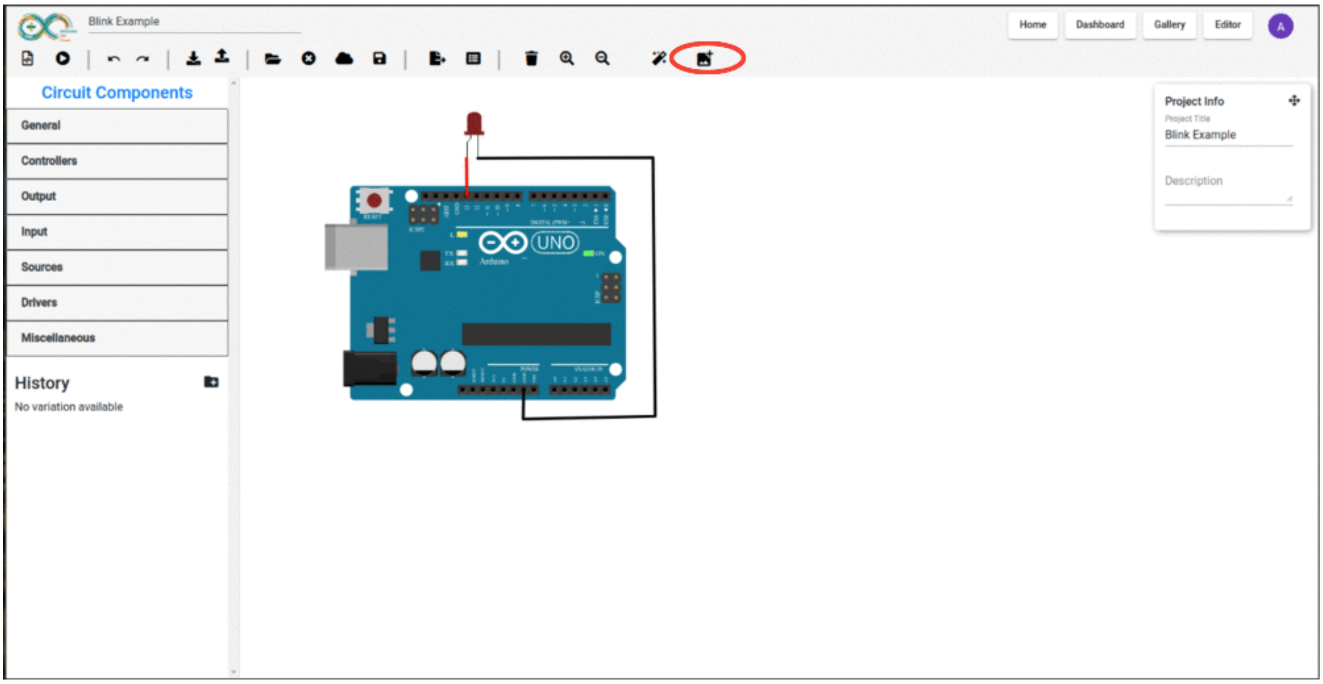


Figure 4.1: Add to Gallery Button

4.2. View circuit on gallery

In the previous model, the circuits were in JSON file and this file was called in the front end to display the circuits. I added all the gallery circuits in the backbend and then I exported this data in migration, I called the API to get the gallery circuits and update in UI, also modified the cards.

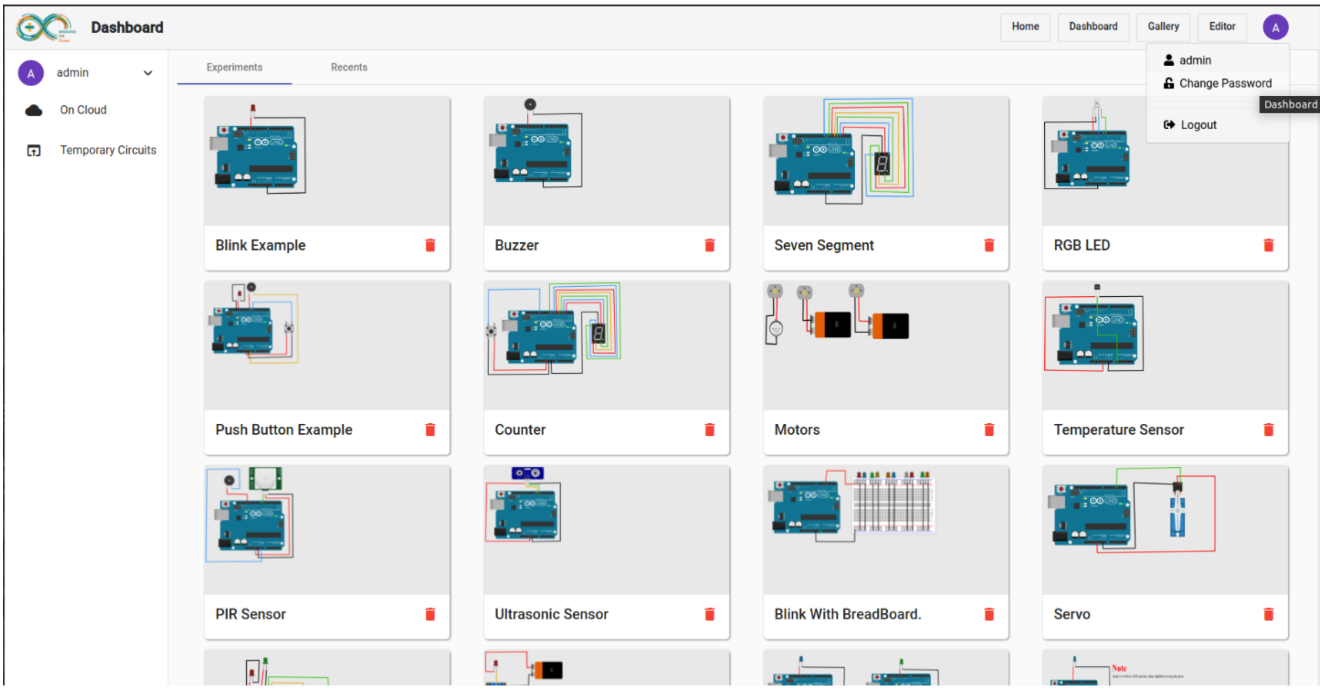
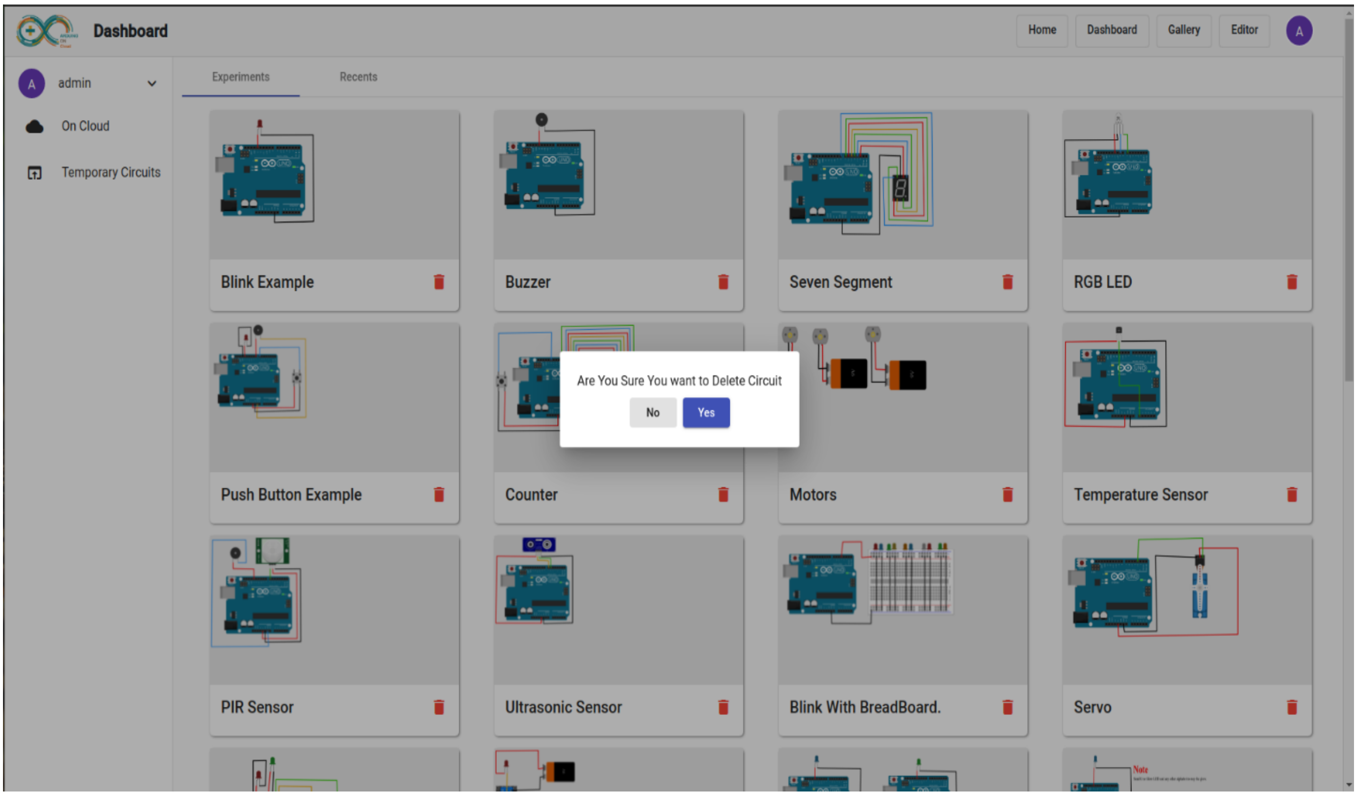


Figure 4.2: View Circuit on gallery.

4.3. Delete circuit from gallery

The gallery is visible to every user but the authority to delete the content is only with the uset with staff role. When we delete an image, it displays the pop-up for the information of the action. After clicking on "Yes", the API for delete will get called.



Are You Sure You want to Delete Circuit

Figure 4.3: Delete Circuit From Gallery

4.4. API Endpoints.

I modified the API's present in the backend including the view circuit API.

Chapter 5

Improvements search functionality.

In the previous existing system this search box consisted of the buttons and the results received where not accurate and the previous version were visible.

5.1 On change Search functionality.

I figured out a npm package named "NG SEARCH FILTER". I modified the previous existing system by removing the search button and when I typed content in the textbox the results is saved where accurate.

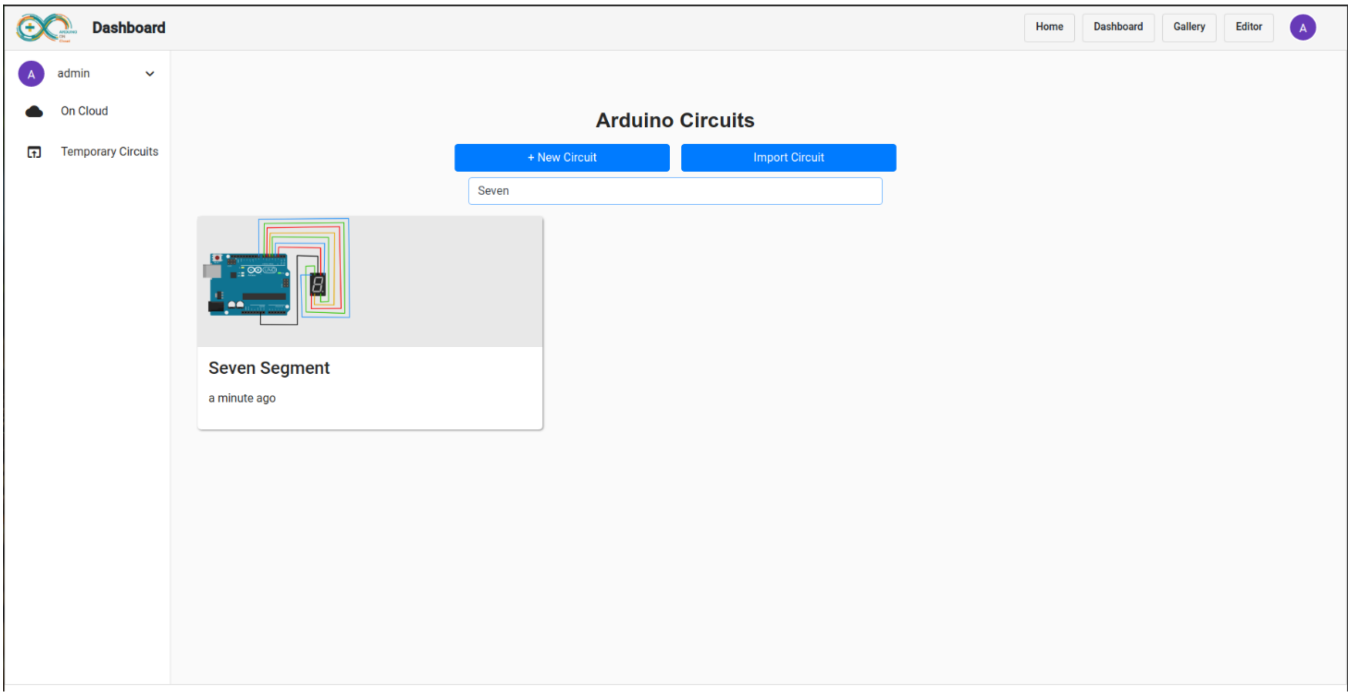


Figure 5.1: Search Functionality for save circuit.

Chapter 6

Some Improvements

In the previous existing system this search box consisted of the buttons and the results received were not accurate and the previous version were visible.

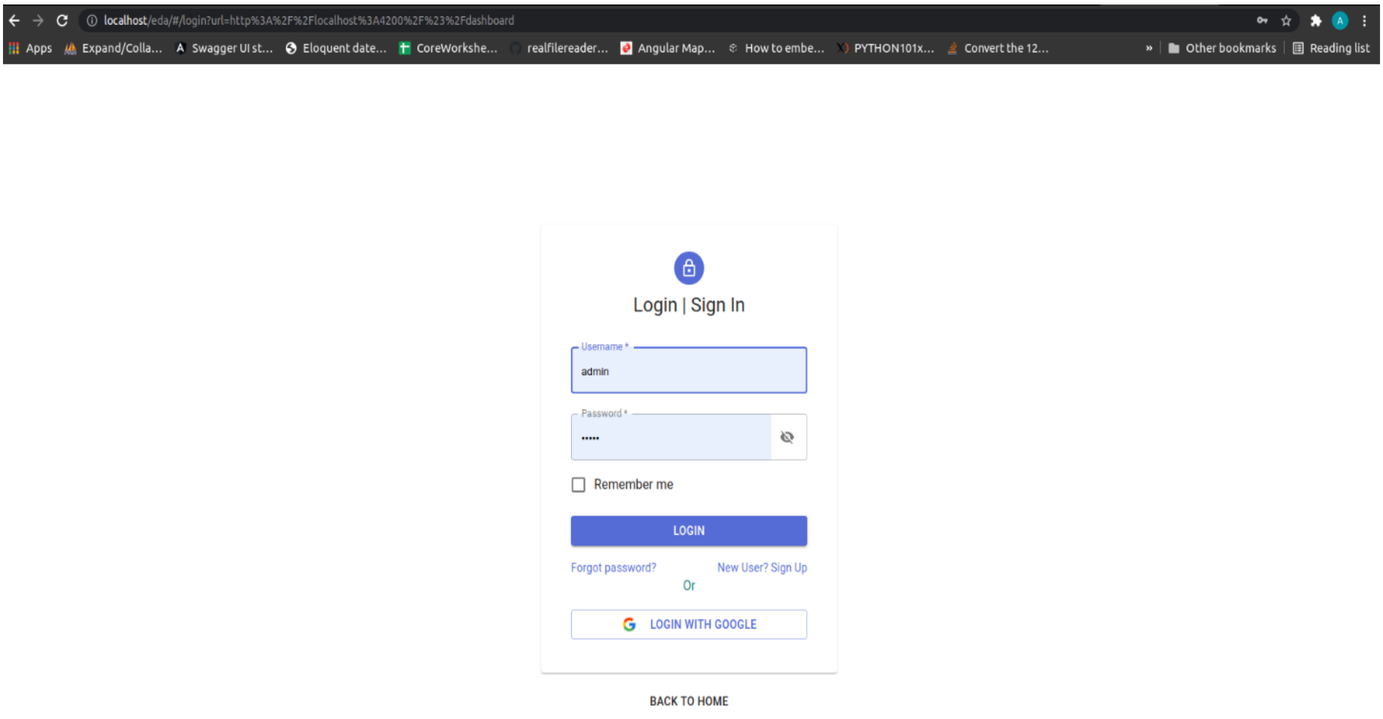
6.1. Show header after login

In the previous system when a user logged in the arduino the header was not displayed. So I modified the code where the user can see the header after logging in.

6.2. Development mode login bug.

when the work in the development mode token is not received while we log in. but this token is present in the production after login in. The main problem faced was when angular ran on development mode, the web pack (disc folder) was not present in the physical mode.

So, When we start with the login process, we can check if the angular project is working with the development as well as production mode. If the development mode is present we can pass the token using URL and in production mode,, this token is hidden.



localhost/eda/#/login?url=http%3A%2F%2Flocalhost%3A4200%2F%23%2Fdashboard

Login | Sign In

Username * admin

Password * •••••

□ Remember me

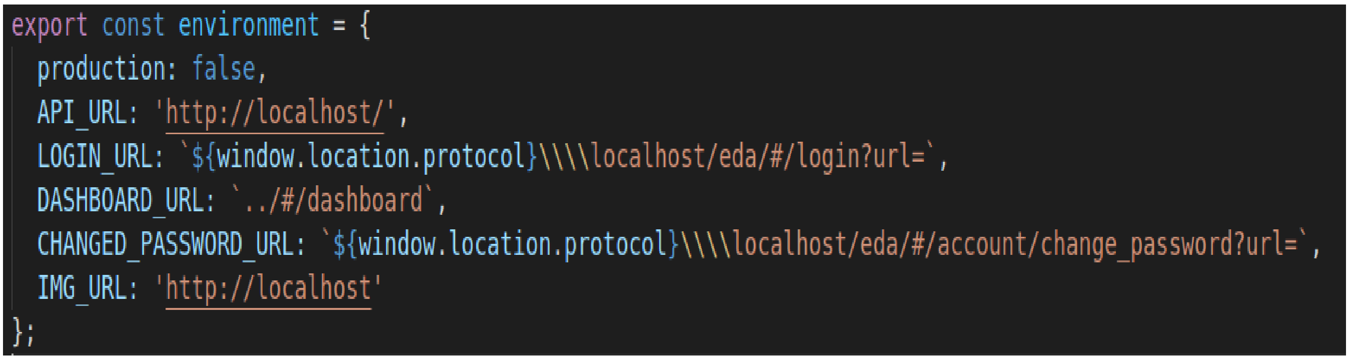
Figure 6.2: Development Mode Login.

6.3. Cloud Circuit save bug

In development mode, when we save circuits, functionalities do not work smoothly displaying errors. The problem occurred due to cors error. I fixed this error to modify the http headers.

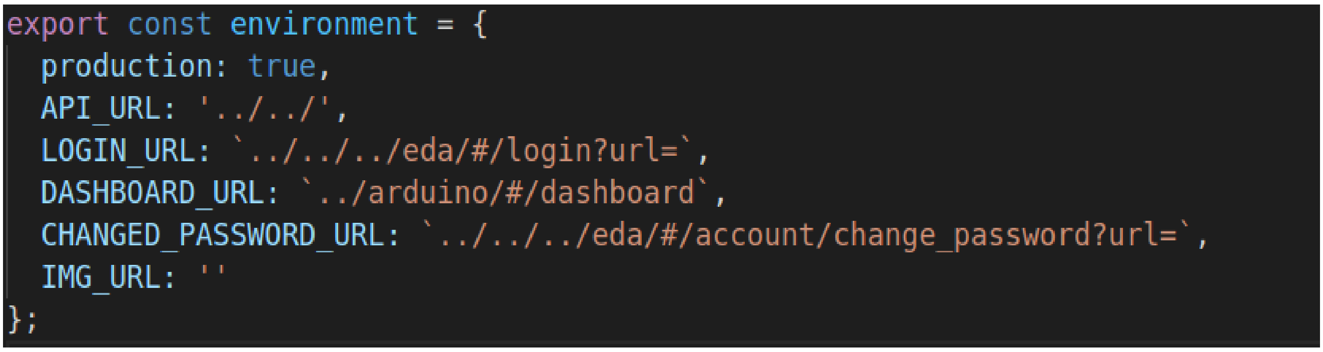
6.4. Change password functionality

This functionality was already present in e-sim, but not in the Arduino cloud. I added the change password functionality in the Arduino cloud. While modifying I only pointed to the URL in production as well as development mode.



```
export const environment = {
  production: false,
  API_URL: 'http://localhost/',
  LOGIN_URL: `${window.location.protocol}\\\\localhost/eda/#/login?url=`,
  DASHBOARD_URL: `../#/dashboard`,
  CHANGED_PASSWORD_URL: `${window.location.protocol}\\\\localhost/eda/#/account/change_password?url=`,
  IMG_URL: 'http://localhost'
};
```

Figure 6.4.1:Environment Urls in Development Mode.



```
export const environment = {
  production: true,
  API_URL: '../../',
  LOGIN_URL: `../../../eda/#/login?url=`,
  DASHBOARD_URL: `../arduino/#/dashboard`,
  CHANGED_PASSWORD_URL: `../../../eda/#/account/change_password?url=`,
  IMG_URL: ''
};
```

Figure 6.4.2:Environment Urls in Production Mode.

Chapter 7

DHT11–Temperature and Humidity Sensor

This is a commonly used temperature and humidity sensor that comes with a dedicate NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data.

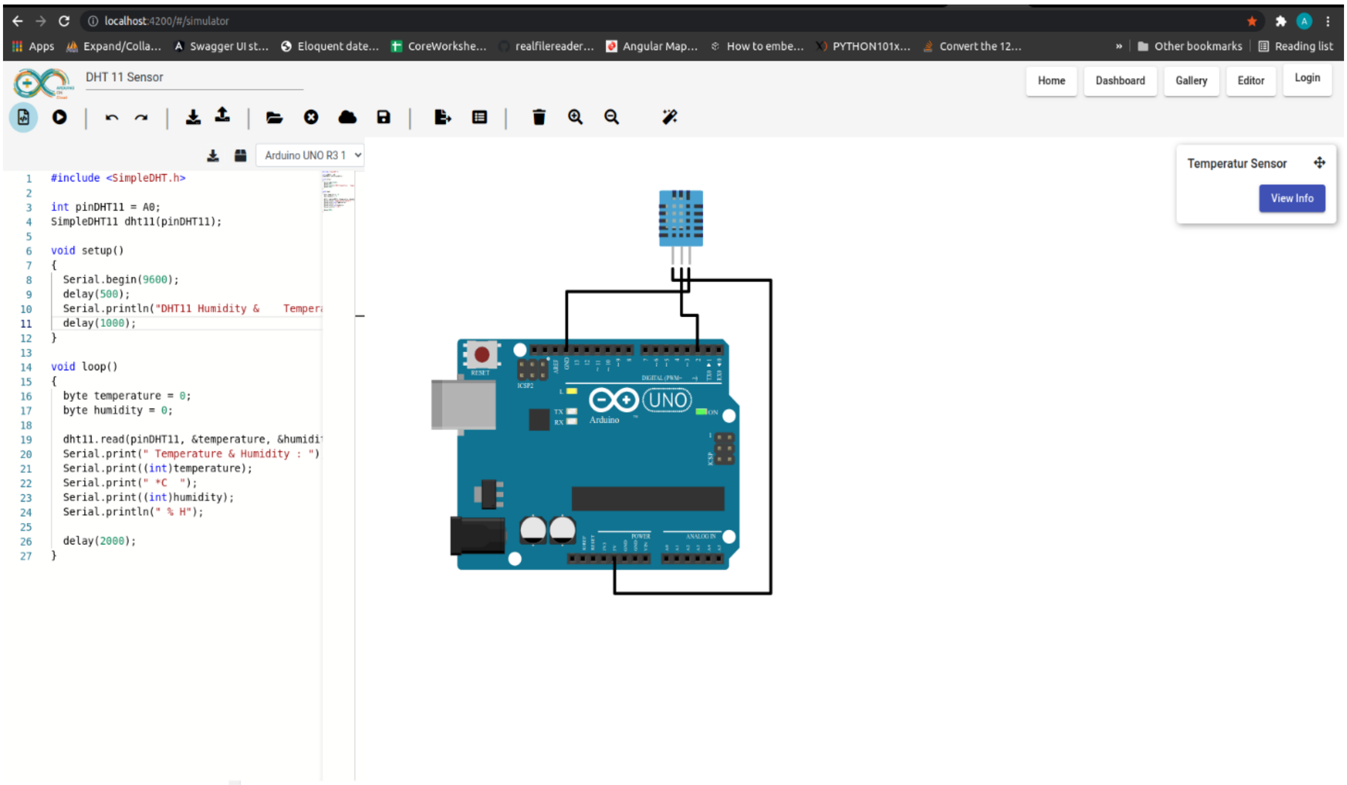
Specifications:-
Operating voltage:3.5 to 5.5 v
Operating current:0.3 mA (measuring) 60uA

Power and Pin -
DHT11's power supply is 3-5.5V DC. When power is supplied to the sensor, do not send any instruction to the sensor within one second in order to pass the unstable status. One capacitor valued 100nF can be added between VDD and GND for power filtering.

Communication Process -
Single-bus data format is used for communication and synchronization between MCU and DHT11 sensor. One communication process is about 4ms. Data consists of decimal and integral parts. A complete data transmission is 40 bit, and the sensor sends higher data bits first. Data format: 8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data + 8 bit checksum. If the data transmission is right, the check-sum should be the last 8bit of "8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data".

Initially I gathered the information about the DHT11 sensor followed by the creation of an SVG file. I connected the DHT11 with an Arduino, using a simple DHT11 library, but when simulation of the program is done moisture and humidity is detected as 0, also accurate output is not detected. I am still working on this to get the accurate result.

Simple DHT Library -
SimpleDHT sensor library is compatible with multiple low-cost temperatures and humidity sensors like DHT11 and DHT22



```
#include <SimpleDHT.h>

int pinDHT11 = A0;
SimpleDHT11 dht11(pinDHT11);

void setup()
{
  Serial.begin(9600);
  delay(500);
  Serial.println("DHT11 Humidity &    Tempera
  delay(1000);
}

void loop()
{
  byte temperature = 0;
  byte humidity = 0;

  dht11.read(pinDHT11, &temperature, &humidi
  Serial.print(" Temperature & Humidity : ")
  Serial.print((int)temperature);
  Serial.print(" *C  ");
  Serial.print((int)humidity);
  Serial.println(" % H");

  delay(2000);
}
```

Figure 7.1:DHT 11 Connection with arduino

Chapter 8

Conclusion

This project was very interesting for me to work on as there were various new challenges. I had a very great time working over it and also learnt various new concepts which helped me to improve my skills and learn new technologies. I would like to thank all the faculty involved for guiding me throughout the project.

Chapter 9

Future Work

- The current system is not responsible so, in future we can work on responsive web pages.
- working with DHT 11 sensor to arduino cloud.
- Refactoring of application.
- Fixing bug, users are able to delete/interact with elements even when simulation is running.
- Adding more sensors and components.

References

[1] https://esim-cloud.readthedocs.io/en/latest/overview/index.html#arduino-on-cloud

[2] https://update.angular.io/?v=7.0-11.0

[3] https://www.npmjs.com/package/ng2-search-filter

[4] https://components101.com/sensors/dht11-temperature-sensor