WEB701 Blogs

Carlos Amos

Contents

Week 1 2
Week 2 2
Week 3 2
Week 5 3
Week 6 3
Week 10 3
Week 11 4
Week 12 6
Week 13 7
Week 14 7
Week 15 7
Week 16 7
WEB701 Summary 10

Carlos Amos

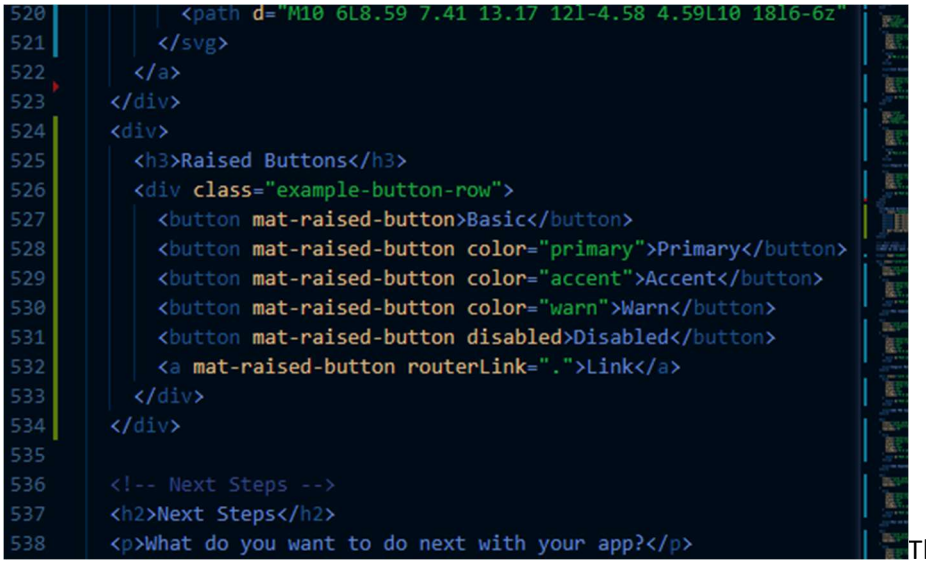
Week 1

For this week, we were given an overview of the contents for the course. We got into groups and looked at the project brief.

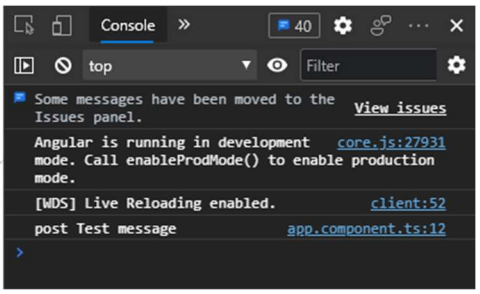
For the project, we must develop a website for a charity of our choice which could also be a fictional charity. The charity I decided on is a fictional charity which I have not decided a name on yet, is a charity which offers supplies to students/families within the community. The supplies will be available to purchase using tokens which will be supplied to them.

Week 2

For this week we started on a tutorial on angular and MongoDB. I have never used angular before so going into this tutorial will be a new experience for me. For this week I managed to get the website up and running and implement a few UI features.



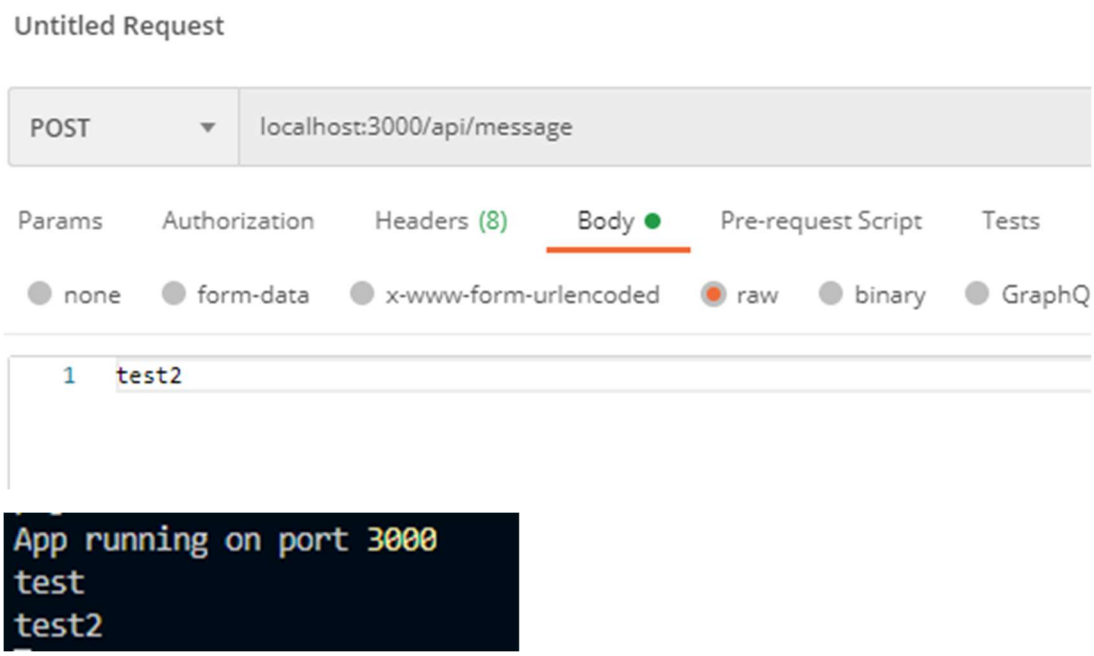
This week, I made it up to the tutorial where we must create a button and text field and when the button is pressed, it will post a message to the console. I successfully added these features, and they were able to post a message to the console.



Week 3

For this week I continued working on the tutorial from last week. I managed to get a working express backend up and running and installed MongoDB as it was needed for the tutorial.

I managed to get a working post where it would send the message to the backend. The backend successfully received the message I sent, and it displayed it on the console.



I then got the MongoDB server setup, and the backend was able to communicate with it. Any messages I send from the angular website get sent through the express backend to the MongoDB database.

Week 5

As milestone one for the project was approaching, I started work on my document for the website and came up with a name for the charity. SuppliesForSchool. As I have created a website in the previous WEB course WEB601, I decided to use the document from that project as a template from this project. The document would contain the website brief, website goals, user experience, site content, visual design, CRUD analysis and the user stories.

Week 6

I finished work on the document and created a prototype backend but as I left the backend to last minute, I did not add anything to it.

Week 10

I started work on milestone two and decided on the two frameworks I am going to use which are Angular and Vue. I decided on Angular as it was the first framework, we looked at earlier in the course. I also decided on Vue as there was some material for Vue in the class Moodle. As it has been a while since I have used Angular, I will most likely have to relearn it and may have to follow the tutorials again.

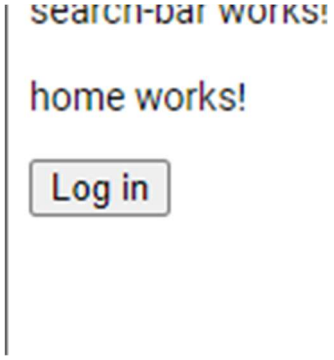
So far, I have created just the UI for the Angular website and have decided to start learning Vue as it is a very new framework to me.

At the end of this week, I have created a working Vue website which only contains the routes and pages but learning Vue, I found it a lot easier to learn then Angular. But I am unsure on which framework to go with now as Vue does not have a lot in terms of templates I can use for the UI elements of the website. I did discover Bootstrap Vue which I have implemented into the website.

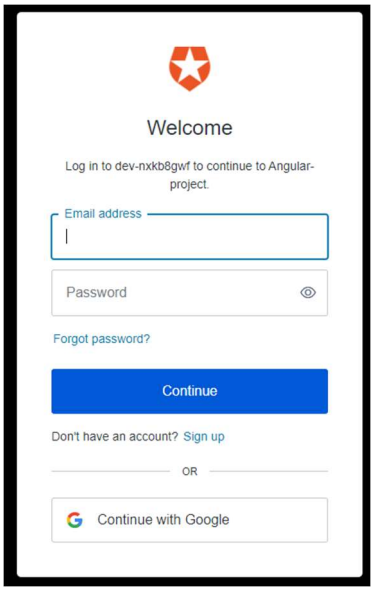
Week 11

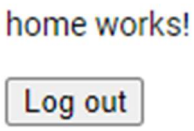
I continued work on milestone two and may have not anticipated the amount of work needed for this assignment. While working on the prototype websites, I decided on working on one framework for a couple of hours, and then switching to the other framework to try and keep what I need to do fresh in my memory.

For the login system for the websites, I decided on implementing a third party auth service called Auth0 which I decided to implement into both my websites. I implemented it into Angular and it is successfully working and can login and register but I am unsure of how to get users to register as a donator or beneficiary and have it in the system as the user being part of one of those parties.



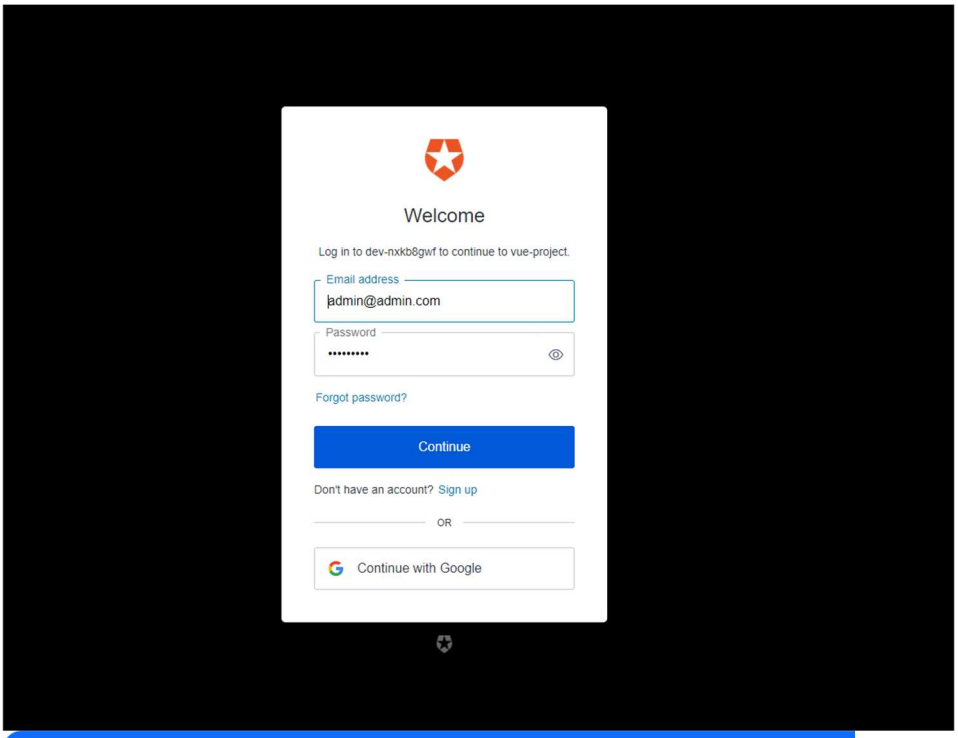
Login button on the home page used to open Auth0 login





Once the user is logged in, the button changes to log out

I have implemented the login for the Vue app but did run into a problem where the authentication wasn't working as when logged in, the log out button wouldn't show but made changes to my code using code within the Auth0 website which fixed it. I can now login using Auth0 on the Vue app.



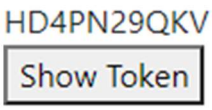
Created a simple prototype profile page

With my recent thoughts on how to register as a donator or beneficiary, I decided to change my mind with the login authentication and have decided to use Firebase instead which features an authentication system but also features a database which I could use for the database of my website.

So far, I have created a Firebase database and a backend which is able to create, post and update listing in the Firebase database from the backend. At this stage, I may not complete all the requested prototype features for the website and have decided to get the features working on Vue first before implementing them into Angular.

On Vue, I have got prototype store working which is able to view all the listings of products within the database and am able to register new products, currently the login is not fully implemented so anyone can register a product. I have also got a prototype token system where the user can purchase a product and when the user presses purchase, it generates a random token for the user and displays it to the user.

```
export default {
  data: function() {
    return {
      tokenLength: 10,
      optiondata: [
        {
          //Generator parameters
          name: 'uppercase',
          status: true,
          chars: 'ABCDEFGHJKLMNOPQRSTUVWXYZ'
        },
        {
          name: 'numbers',
          status: true,
          chars: '0123456789'
        }
      ]
    }
  },
  computed: {
    charset() {
      return [...this.optiondata]
            .map(element => {
              if(element.status === true)
                return element.chars;
            }).join('');
    },
    //Generate token every time its called. Called on page startup
    generatedToken() {
      this.refreshToken;
      return [...window.crypto.getRandomValues(new Uint32Array(this.tokenLength))]
            .map(value => this.charset[value % this.charset.length])
            .join('');
    }
  }
```



Tokens is randomly generated when the show token button is pressed.

At the hand-in stage, I got at a prototype of the requested features for the Vue website working but for the Angular website, the only working feature was the login system.

Week 12

Right after handing in milestone 2, I decided to get a start on the website straight away as I did not want to end up in the same situation. I decided on keeping most of the features made in milestone 2 for milestone 3. The system I want to change is the token system as instead of giving the user a token at purchase, the user will be able to acquire tokens if they are a beneficiary and use the tokens to purchase the product of which I am pretty sure is a requested feature instead of the other token system.

So far, I have got all the pages, routes and navigation set up.

Week 13

This week I worked on the listing page in the store where it displays the listings of products. I have used the prototype from milestone 2 for this project as I was happy with that prototype. This week I want to work on getting all the features for this system finished.

I have implemented the store and am able to view details of an individual product when pressing the view button. I have decided to start work on the login system of the website.

Week 14

This week I continued work on the login/register component of the website. Currently, I have the controller, route and model set up in the backend and can register and get users in postman.

I have decided on ditching any third-party authentication system and have decided to implement my own. I have implemented a jwt service for when a user logins in. The register forms are fully working, and the login form is also working but I have run into an issue where the app is unable to authenticate whether the user is logged in or not. The problem I've been having is the app can successfully check the user and retrieve the authentication token, which is stored as a cookie, but the app gets an authorization error trying to retrieve the user details using the token which has been stored.

Week 15

I have continued to work on fixing the problem from last week but am still unable to resolve the issue. I have tried changing the code within the backend to retrieve the token by putting it in the body of the post message but still get the same unauthorized error. As I feel I am wasting time trying to fix this error, I have decided to work on the token system. I have currently implemented a way of storing the user by storing their email in the local storage and using an if statement to display the profile and log out button on the app. The if statement works by first getting the email stored from the local storage, and if there is an email stored there, it will display the log out and profile button.

Week 16

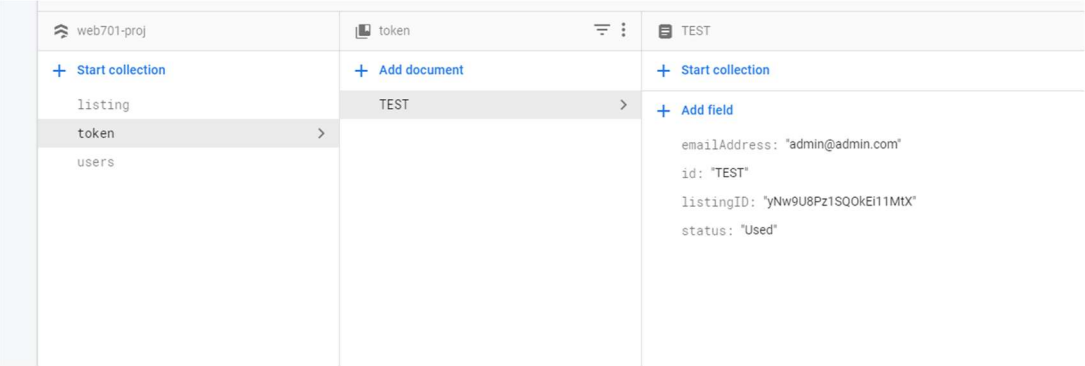
This is the final week of the course and have decided on the final feature I am going to implement. If I cannot get the authorization token system to work on the website, I may continue with storing the email in local storage and continue checking to see if the user is logged in that way.

I have the token system setup and ready so an admin can generate the tokens. I do need to work on the acquisition of the tokens from a beneficiary, but the plan is to have it on the profile page where the user is informed they have tokens they can acquire, and when they press acquire on the tokens, it will update in the database that the token has been claimed, the token can then be used to purchase a product.

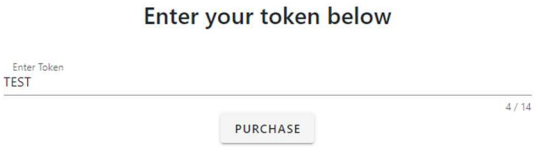
Also created an admin section to the app that can only be accessed if the accounts party is Admin. In the admin app, it will currently only be used to enter a desired account email address and generate a token for that email address.

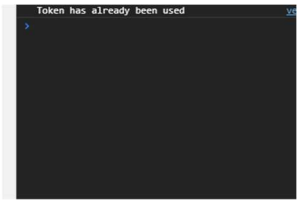
So far, I have got a working token system working where a beneficiary can request a token, the beneficiary can view the token in the profile page, that token can then be used to purchase a package/product from the store and the token will be updated to ‘used’ once product is purchased that was it cannot be used again.

Added test token to test the token system that checks if token has been used or not.

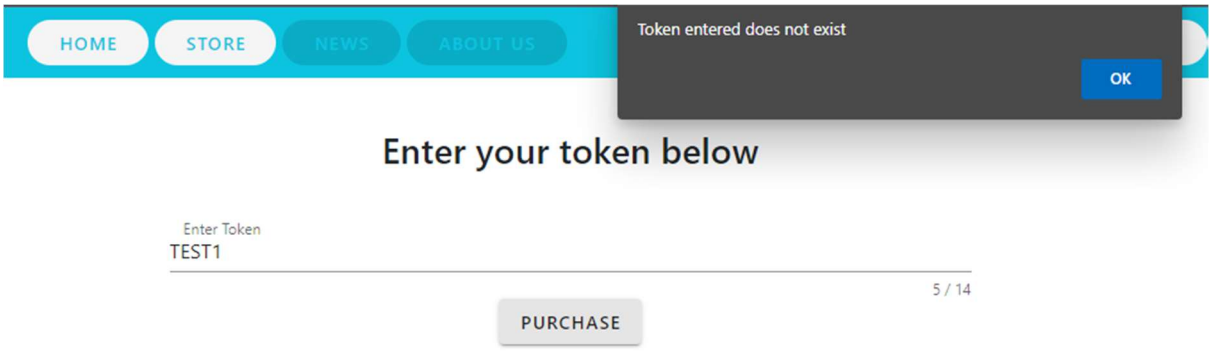


App shows that token has been taken





App shows that token entered doesn’t exist.

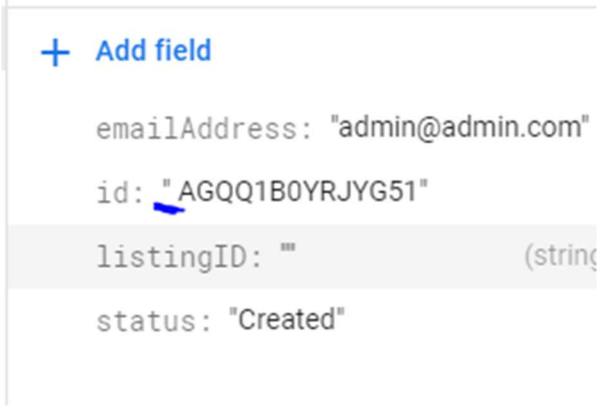


Code for the system.

```
    const data = {
      id: this.id,
      emailAddress: localStorage.getItem('userAccount'),
      listingID: localStorage.getItem('userAccount'),
      status: 'Used'
    }
    Vue.axios.get(`http://localhost:4200/api/token/${data.id}`)
    .then((res) => {
        if(res.data.status == 'Used')
        {
          console.log('Token has already been used')
        } else {
          console.log('Success')
        }
    }).catch(() => {
      alert("Token entered does not exist")
    })
  }
}
}
</script>

<style>
```

I did run into a small issue with the token system where the entered token wasn't recognised in the database for any of the automatically generated which would result in a token does not exist error even though the token was valid. The problem with the issue was the automatic token generator where the generator would put a space at the beginning of the token.



This issue was resolved by adding a space in front of the token in the search feature.

```
}
var checktoken = " " + data.id
Vue.axios.get(`http://localhost:4200/api/token/${checktoken}`)
.then((res) => {
```

It is time to hand in the project. I have added nearly all the features requested but the features I did not implement was a feature to edit and delete an existing listing. I also did not manage to get the login auth working but decided to stick with storing the user information in the local storage and deleting it when logging out.

Overall, I am reasonably happy with what I got done for milestone three and feel I learnt a lot from this project. If I were to do a future project involving a web app, I feel like Vue would be my first choice of framework.

WEB701 Summary

Overall, I feel like I learnt a lot of information coming out of this course. I enjoyed learning Vue and will most definitely continue using it outside of the course. Learning Angular, I did not enjoy so much as I found it a little bit confusing relearning it for milestone two.