Josh Munro

[Company name]  [Company address]

Web701 Evaluate part2

Table of Contents

[Introduction 2](#_Toc73394199)

[GitHub repo 2](#_Toc73394200)

[Written Report 3](#_Toc73394201)

[Features used in my website. 3](#_Toc73394202)

[Frameworks I am using. 4](#_Toc73394203)

[Charity members/beneficiaries register, login and administrate their own account. 5](#_Toc73394204)

[Angular 5](#_Toc73394205)

[Angular Registration process 5](#_Toc73394206)

[Angular Login process 7](#_Toc73394207)

[Angular Administer own accounts. 8](#_Toc73394208)

[Compared to Vuejs Registration, login process administer accounts 10](#_Toc73394209)

[Users administrate their accounts. 12](#_Toc73394210)

[Admin/Members Register items to website and beneficiaries use system to acquire tokens. 13](#_Toc73394211)

[Angular 13](#_Toc73394212)

[Use system to acquire tokens. 14](#_Toc73394213)

[Compared to Vuejs registering and requiring tokens 15](#_Toc73394214)

[Comparison 17](#_Toc73394215)

[Beneficiaries use system to acquire tokens. 18](#_Toc73394216)

[Angular prototype 18](#_Toc73394217)

[Compared to Vuejs prototype 20](#_Toc73394218)

[Interactive elements that engage the website user 21](#_Toc73394219)

[Angular Interactive elements that engage the website user 21](#_Toc73394220)

[Vuejs Interactive elements that engage the website user. 23](#_Toc73394221)

[Compared to Vuejs 25](#_Toc73394222)

[Comparison 26](#_Toc73394223)

[Store and retrieve data from an API. 27](#_Toc73394224)

[Angular Store and retrieve data from an API. 27](#_Toc73394225)

[Vuejs Store and retrieve data from an API. 30](#_Toc73394226)

[Recommendation of framework to use. 32](#_Toc73394227)

[Pros and Cons of Angular 32](#_Toc73394228)

[My opinion and experiences 33](#_Toc73394229)

[Table of contents 34](#_Toc73394230)

# Introduction

Welcome to my assessment two stage two assessment. In this assessment I have been creating components and proof of concept code that compares how two different web frameworks work. The two frameworks I have studies and compared are, Angular, and Vuejs.

The assessments discuss and compares how these frameworks can do the following in a proof-of-concept manner.

* Beneficiaries and charity members can log in register and administrate their accounts.
* Interface for charity members to upload their services, in my case the charity members create soup items, with a name and description, the items can also come with a promo code.
* Interactive elements between Angular and Vuejs
* Charity members can accept a token. The beneficiaries choose what item they want and enter the promo code as a form of transaction.
* Store and retrieve information from an API

In my web apps I have created two different frontends that use one backend restful API.

**The name of the Angular App is NgApp.**

**The name of the Vuejs app is Vue-router.**

You can access the web app and restful API through the git repository I have created.

# GitHub repo

**https://github.com/Josh-Munro/Auth.git**

# Written Report

A web framework allows for web apps to be developed, published and updated. A web framework is fundamentally a library that allows for faster and easier dynamic web development (*What Is a Web Framework? - Evolve Evolve*, n.d.). It allows for scalable dynamic components to be created. Along with scalable and maintainable web applications to be developed (*What Is a Web Framework? - Evolve Evolve*, n.d.).

There are a few similarities and functionalities shared between web frameworks.

* URL routing. The application can open and switch to different view and components on the webpage with any loading of code (*Framework Main Features - Learn Web Development | MDN*, n.d.).
* Supports HTML, CSS, XML, JSON (*Framework Main Features - Learn Web Development | MDN*, n.d.).
* Database orientated. Configure and add database connections.
* Security against hacking techniques such as SQL injection, cross site scripting.
* API development and configuration from server to client.
* MVC structural pattern
* Use underlying data resources API (*What Is a Web Framework? - Evolve Evolve*, n.d.)
* Re-usable code (components)
* Most frameworks are open source (free) (*What Is a Web Framework? - Evolve Evolve*, n.d.)
* States – components have states – states can effect how component rendered
* Events – event handling (*Framework Main Features - Learn Web Development | MDN*, n.d.)
* Handling dependencies - (*Framework Main Features - Learn Web Development | MDN*, n.d.)
* Nested components - (*Framework Main Features - Learn Web Development | MDN*, n.d.)

## Features used in my website.

The features identified above that I have used in the web development of my Angular and Vuejs frameworks are…

URL routing. Both frameworks use some form of routing either using Router from Vuejs or Angular.

Another feature used is the use of components. Both frameworks I have used are component based and use components to build a web application.

Both of my applications have a backend restful API to connect to a database and retrieve and post information. Hence both frameworks are database orientated and allow integration of APU’s.

Both frameworks used are open source.

Both framework use JSON. Responses and request come in the form of JSON.

Most web frameworks use the MVC (model view controller) architecture system to separate data model from the logic. This benefits web development as it promotes code reusability, this also helps for different views such as web pages to be viewed. (“Web Framework,” 2021).

## Frameworks I am using.

The web frameworks that I will be comparing and exploring will be Angular and Vuejs.

**Angular** is a framework that allows developers to create single page applications. The languages used in the framework is Typescript and HTML.

A big part of Angular is that it allows reusable components to be created and made for building the web application.

An awesome feature of Angular is the built in CLI. The CLI has some amazing features which make web development easier and faster. There are commands such as ng g c Navbar. This command allows a component of the name Navbar to be created. This includes a new folder containing the typescript JavaScript, html and CSS files with necessary setup code and imports.

The other framework is **Vuejs**. Vuejs is a framework primarily focused on user interface. The core of the framework is heavily used on the view layer. View js is built to be simplistic and have a small learning curve for beginners. View js is designed to allow developers to declaratively “render data to the DOM” via using a template tags and html syntax similarities.

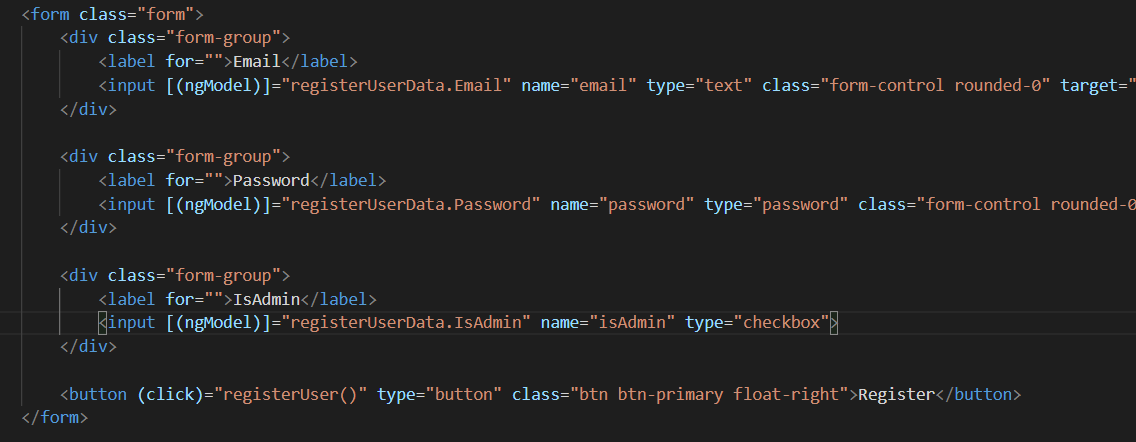
# Charity members/beneficiaries register, login and administrate their own account.

## Angular

In my experience Angular was simple and straightforward compared to Vue. It might be because the process/flow of Angular suits my learning style and how my brain works.

## Angular Registration process

For creating a **registration** using Angular, I had to create a register component. The register component automatically includes an html files, typescript file and a CSS file. The html file is where the elements and building blocks of the component is created. In this instance I created a form using form, div, h1, and input tags. For registration there is an email, password, and admin checkbox field (Codevolution, 2018).



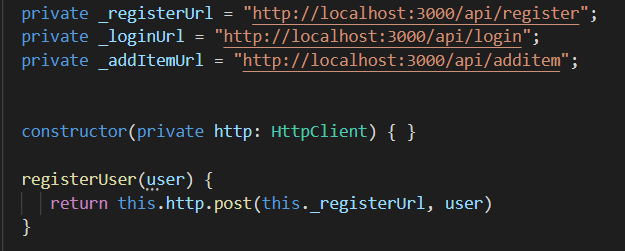
To pass the data from Angular form component to the API, I used ngmodel. I created an object/array from the typescript file initially which included the necessary fields for the form and API. The object is called registerUserData. In the form the registerUserData.(FieldName) is assigned to the user input (Codevolution, 2018).



In the button component the information is passed to the auth service. Angular injects the auth service into the constructor.

The auth service is separate to the register component. The auth service then passes the information to the API (Codevolution, 2018).





I really like the flow and logic of this system. There is separation between the form, the logic and passing through data to the API.



## Angular Login process

The login system is pretty like the registration process. However, in the login process the API acts differently to registration API.



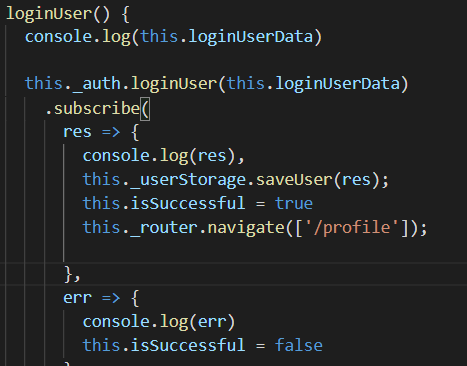
This is the Login html component file. Notice each field is assigned to a user object field. Email input field for instance uses ngmodel to store the input from the user to the object field.



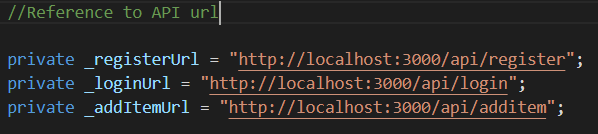
The loginUser function is called when the user clicks on the login button.

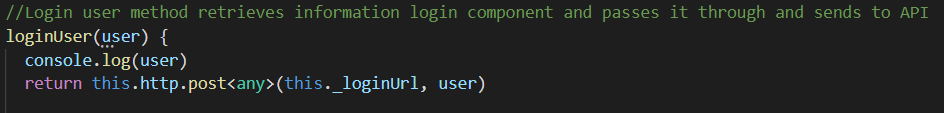


The login typescript file is passing AuthService and UserStorageService to the constructor as these are needed further down the file.



The information from the user object is passed to the \_authService, from there the AuthService will pass the information to the API.

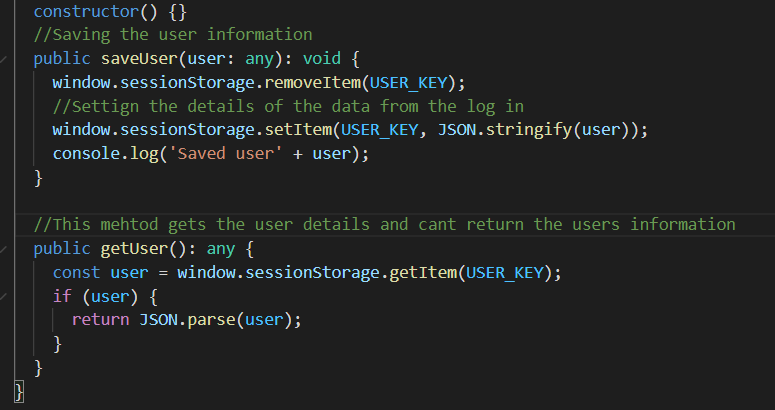




## Angular Administer own accounts.

As there are different roles/types of users for the charity application, I would need to have different users displaying different information on the screen.

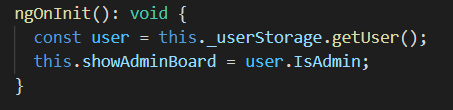
For example, the Angular application will need to display components depending on the role of the user. In Angular I did this by firstly creating a **user storage service.** The purpose of the storage is to store the temporarily logged in user information. In doing this if the isAdmin field is equal to true then some certain components will display on the admin dashboard.



Then I injected the user storage service into the login component, because I need to save the user information.



To use the user information from the database all you need to do is call the getUser() method from the user storage service.

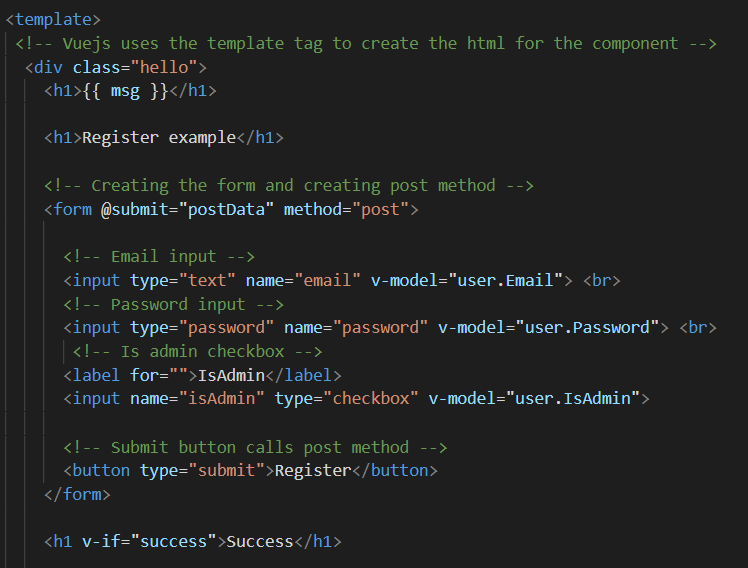


Then set a variable to user.IsAdmin for specific components to display depending if the user is an admin or not.

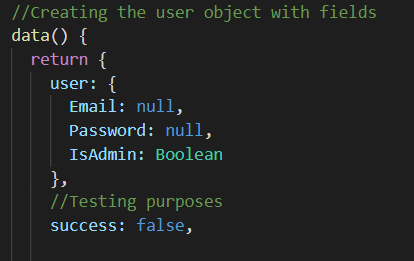
## Compared to Vuejs Registration, login process administer accounts

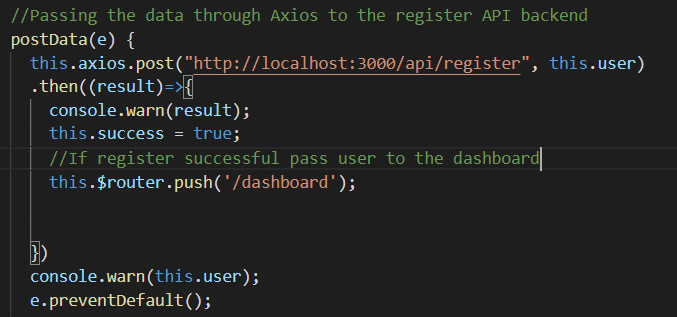
However, compared to Angular Vuejs, has a similar approach fundamentally but has a few interesting way of passing the data.

Vue has a data function which returns values. In this function I created the user object, with the corresponding fields. In the form html, you use v-model=”user.Email” to pass data from form to the logic area.



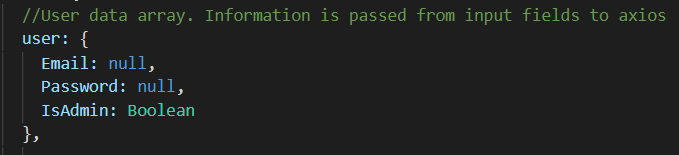
Next Vue uses axios as a way to pass the form data to the API. In the postData function you use axios.post(‘api URL’) passing this.user as the object/data. Whereas Angular uses an extra step and injects the \_auth service into the constructor.

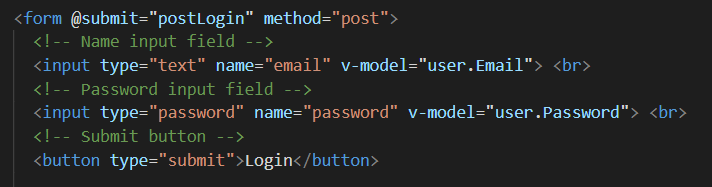




This system uses axios instead of a service. However, I enjoy using Angular more, because Vue the style and architecture of Vue is confusing and messier and makes less logical sense in my opinion.

The process is the same for login, however, the information is being passed to the LOGIN api not the REGISTER api. Except for the isAdmin user field.





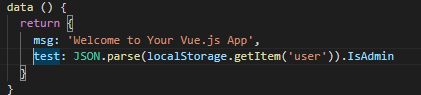
## Users administrate their accounts.

In Vuejs the way to store temporary user information is to use this method shown below.

localStorage.setItem('user', JSON.stringify(result));

Angular uses window.sessionStorage, whereas Vuejs is simpler and does it by using localStorage. Angular also requires a USER\_KEY variable to be passed, but from the tutorials it seems that Vuejs does not need this. Both frameworks stringy the JSON result back (*Vue.Js JWT Authentication with Vuex and Vue Router - BezKoder*, n.d.).

Being able to set item means that the framework is also able to get the item using this line of code.



This allows for variables from other components to be set based on the information stored from the login API callback.

This is similar to the Angular app, however, the Angular sets the getitem in the service, whereas here in Vuejs I directly called the method. The look and feel of both frameworks for this specific functionality is fairly similar.

# Admin/Members Register items to website and beneficiaries use system to acquire tokens.

## Angular

When registering items to the website and API it is very similar to the process of registration. In Angular there are input fields made into the component.



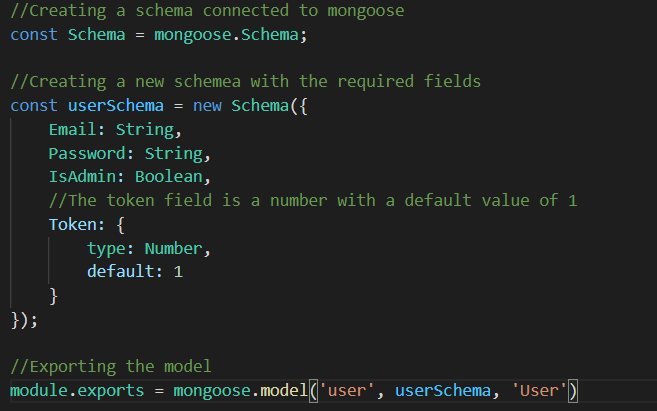
Next the values assigned to the form are submitted and taken to a create Item service. The service then passes the information to the create Item API.



## Use system to acquire tokens.

In how I perceived the assessment. When a beneficiary registers to the website they will gain one token. The beneficiaries can then use this token in transaction further into the future to redeem an item. However, while redeeming an item the beneficiary must also have access to the promotional code. The promotional code is an extra step/security reason for items to be claimed.

In the Angular project when the user registers the user model will automatically fill in the token field with 1 token value. This is shown in the user model in the server (backend).



The client side of Angular then only passes the information of email and password, since the default value for token is set.

## Compared to Vuejs registering and requiring tokens

Similarly, to the registration process on Vue the forms collect the values are passed through a user object containing the correct fields. Next in the methods section of the Vue component, axios is used to post the data to the API. Using this line of code.

 postData(e) {

      this.axios.post("http://localhost:3000/api/register", this.user)

      .then((result)=>{

        console.warn(result);

      })

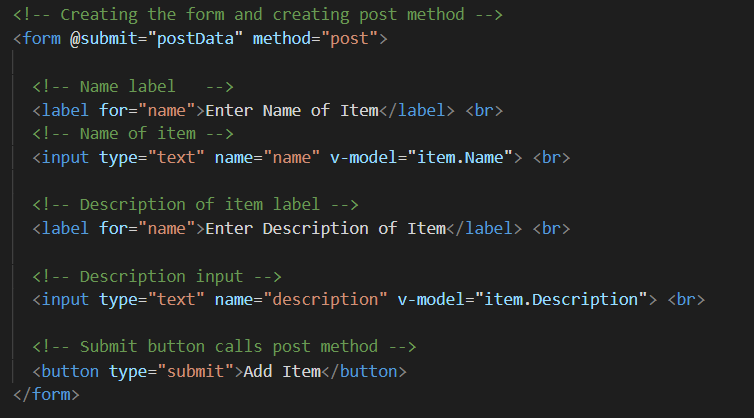
      console.warn(this.user);

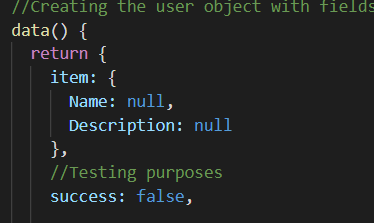
      e.preventDefault();

    }

The post is then taken to the API url, this.user is passed to the API. There will then be a result printed. In this instance, the component may want to set a variable to true or false for a component to display, or for an action to take place. Possibly taking the user to a new route.

The information from the form is passed using v-model. There is also an array/object created that values are set from the user in the form element.





## Comparison

To summarise the differences between the Vuejs and Angular frameworks, Angular uses services, and ngmodel to move data from the form to an array/object with fields. The information is then passed to a service, then the service takes the information to the API.

Vuejs, on the other hand uses Axios to pass the data from the form straight to the API. It also uses v-model to pass input value to an array/object field.

Both Angular and Vuejs have a similar process however, Angular uses typescript so the data array fields must be assigned a type. On the other hand, Vuejs uses javascript, so all the data and fields don’t need to be assigned a type.

Creating methods, are very similar to Angular and Vuejs when doing the assessment. Both use similar syntax to create a method. However, I did notice that Vuejs has a different way to post data when the button is clicked. The form tag already has a submit method, and so when the button is called the submit method is called.

# Beneficiaries use system to acquire tokens.

I had some trouble to work out a way to create a system where users can use tokens to redeem an item. I had an idea where items have a promotional code where the members can select an item and type in a promotional code to redeem the item. However, the API sees if the item exists and if the promotional code is part of the item.

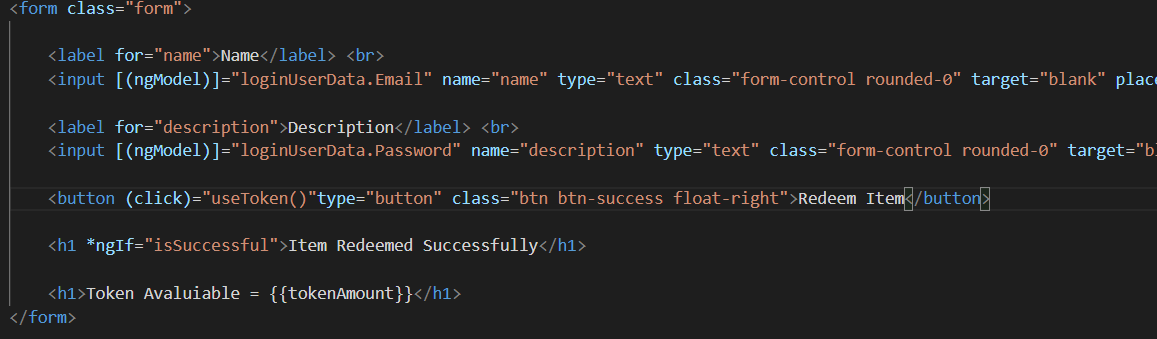
I have a prototype of the concept. The user must type in the Item name, and the promotional code with it. When the users click on the redeem button, if they have a valid number of tokens and the API response was successful then a message in the screen will appear “Item redeemed”.

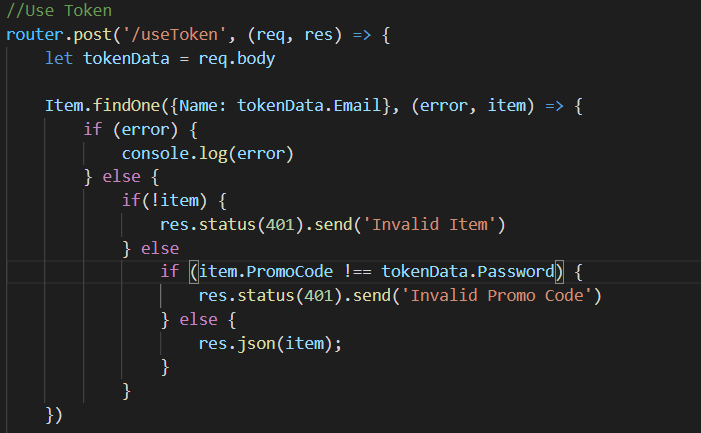
## Angular prototype

In the Angular prototype app, I have used a similar service and API to the login, however if checks if the item name exists and if the promotional code fits with the item. I used a local variable tokenAmount, and if the tokenAmount < 1, then the Item cannot be redeemed.



This can only be a temporary solution as ideally the token amount should be stored in the user database. Since the website page can be refreshed and the token value goes back to its normal value. However, in the example code given, I have demonstrated a way in which the beneficiary’s user types can use their tokens to select an item and have access to the item using a promotional code. This prototype is my way of a interface where users can redeem items.





The problem is that I need to transfer the user data from the database to the component. I will need to call the user storage service and get the token value from the user column. The user must then have enough tokens for a successful transaction.

## Compared to Vuejs prototype

I used the same format and functionality of the login for the promotional code for the Vuejs accepting token interface. In my example there is a template tag with the form elements inside. The data binding using v-model assigns the input value from the user to the data array in the javascript file. In doing that when the button is pressed the data is passed to Axios.

Again, compared to Angular, Vuejs uses the same system of binding array object fields from the javascript/logic, and pushing the information to a service (angular), or in Vuejs Axios. The data is then passed to the API.

Vuejs uses v-model whereas Angular uses ngmodel in a banana in box syntax[()]. Vuejs process for passing data does seem more simple than Angular, along with having similar style of doing things.

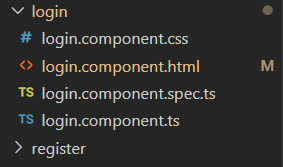
# Interactive elements that engage the website user

## Angular Interactive elements that engage the website user

Interactive elements such as components are an essential part of Angular.

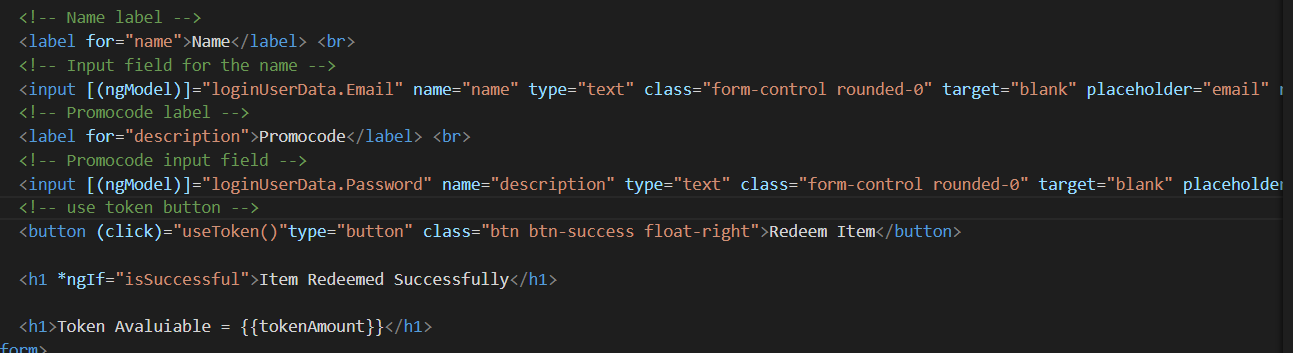
The best way to create a component in Angular is through the Angular cli. To create a component all you need to do is to type ng g c Login. This generates the component with the html file, typescript file.

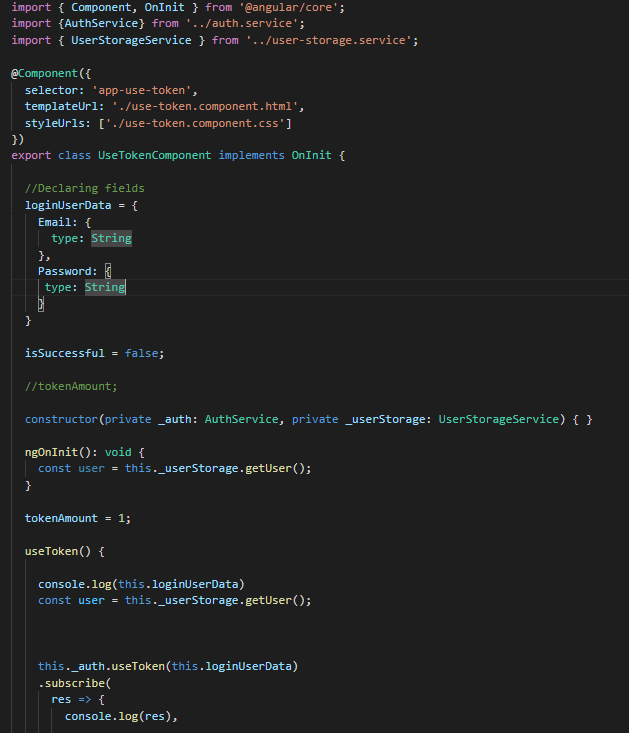
All the files are all set up and ready to go.



For creating.

**Html section of framework**



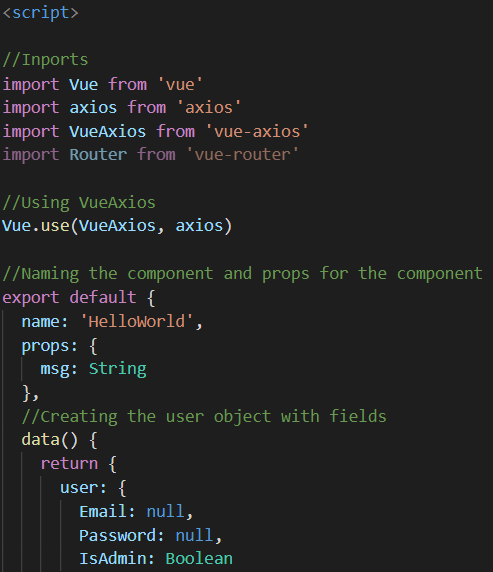


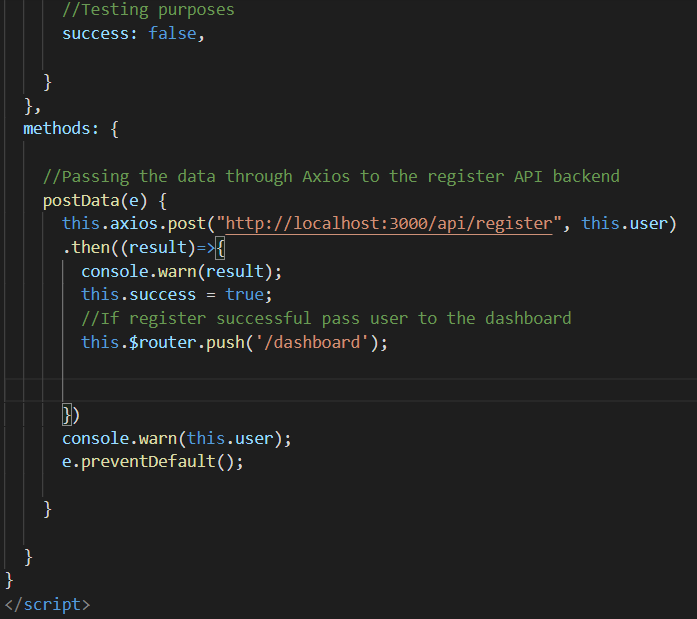
## Vuejs Interactive elements that engage the website user.

In Vuejs there are three things that make up a Component. The first being a template. The template is made using the template tag.



There is a script section that defines the component. There is a script tag as well. Technically all these parts can become separated into different files.





In the template tag there will be the html content of the component such as div and other elements.

In the js script section you must name the component ‘component’. Then there is a data method/attribute which defines the methods and fields for the component.

In order to use the component, you must then use the name of the component as a html tag format and import the component in the script/javascript section of a component.

In angular you use the **\*ng-if** inside a component tag is call if a component should be displayed or not due to conditional logic or Boolean values set to true or false.

## Compared to Vuejs

Compared to Angular, Vuejs has a different way to display interactive components and elements on the web browser.

Vuejs uses template tags to wrap the html content. The logic is defined in a script tag, this is where all the imports, props, and data are returned. There is a method part inside the script tag which handles all the methods and functions, for instance what happens when a button is clicked, and a method is called.

In summary Angular uses a clever CLI to create interactive components that display on the browser, this makes it quick and easy. Whereas Vue can allow for one file to have all the styling, logic and html elements to be displayed.

## Comparison

I like how angular automatically creates the components and automatically generates the necessary files. It is also awesome that the files are all separated and linked to one another. The html o the component does not need any fancy template tag on the outside. It only needs the pure html which is the content of the component.

Unlike Vuejs, you cannot automatically generate components through a command interface. There is the extra template tag that needs to be on the outside of the html component to work. In the logic/javascript of the component you must specifically name the component.

Vuejs also does not use typescript which can be an advantage and disadvantage.

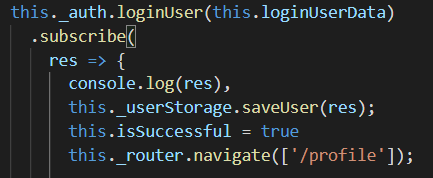
# Store and retrieve data from an API.

## Angular Store and retrieve data from an API.

Both Angular and Vuejs are client-side frameworks and can retrieve, update, and post data into the API and database. As mentioned before in the previous examples Angular and Vue use different ways to pass the information to the API. In my approach for Angular I used Services to pass the data to the API. In Vuejs, I have discussed how I passed information to the API through Axios.

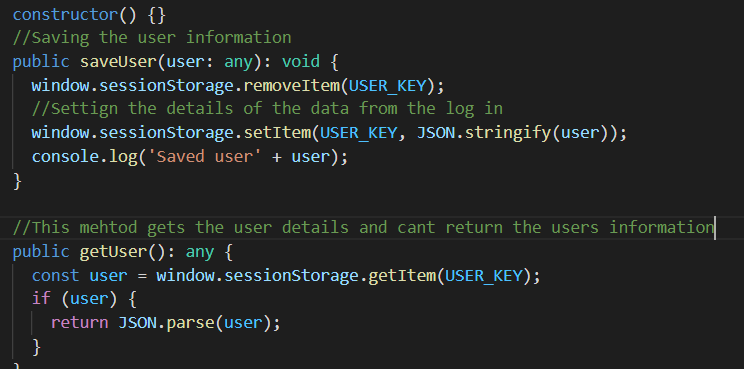
Also, there is similarities to how user information is stored locally and used to manipulate what data is displayed.

In the Angular web application, I created a userStorage service to store the login details of the user on the website at the current time.

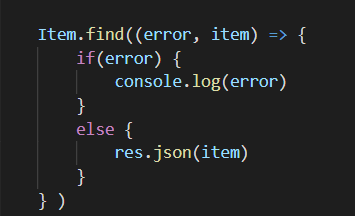


In the userStorage service I used the localstorage to store a temporary token of the current user logged into the system.

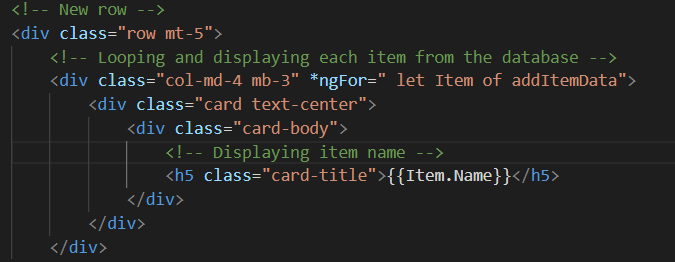
In doing this I am able display information about the logged in user and hide or show components depending on the users role.



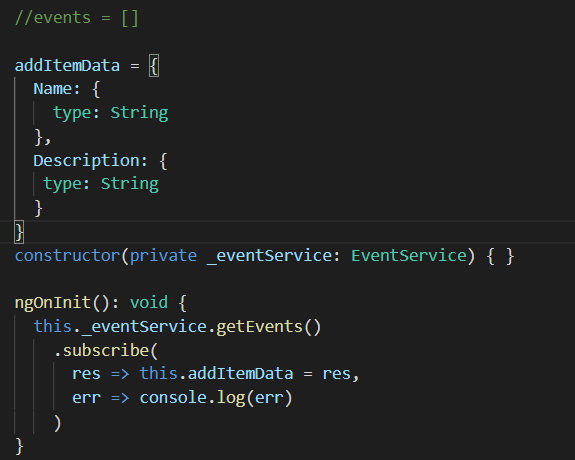
I have also created a backend API to display the items created names into a html viewable component.



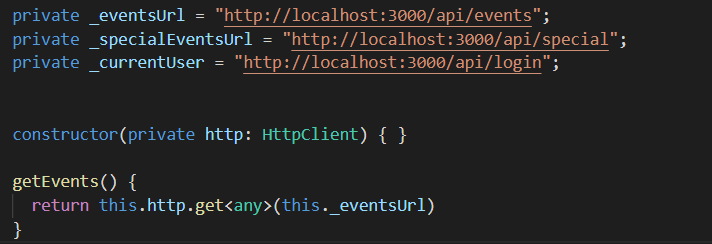
This API is used for both of my front-end frameworks.



This loops each of the items from the database and displays it a new div.



This is the typescript of my items. The file puts the eventservice inside the constructor. The evenservice is called and the getEvents method is called from the even service.



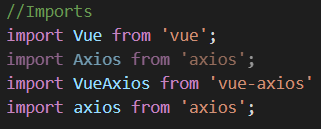
The event service then grabs the data from the database.

## Vuejs Store and retrieve data from an API.

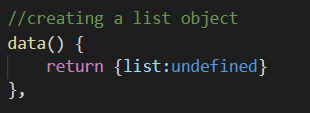
Inserting information using Vuejs has been described in the other process, but as a summary, the mains difference between Vuejs and Angular is that Vuejs uses Axios to quickly pass the information to the API, whereas Angular has a few extra layers/levels. It is technically quicker and easier to pass and retrieve data using Axios from Vuejs, however, I found the process to be more confusing and messier. I rather used a few more layers and steps to make my code cleaner.

Retrieving data from the API with Vue is quite different to the approach Angular took.

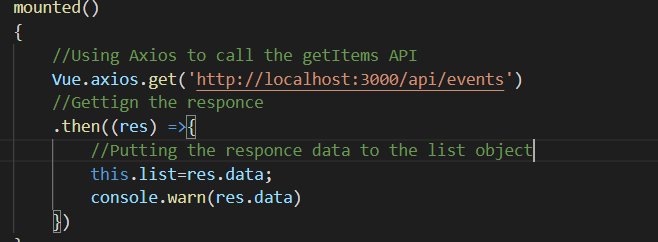
First you must import Axios, Vue, and Vue-Axios.



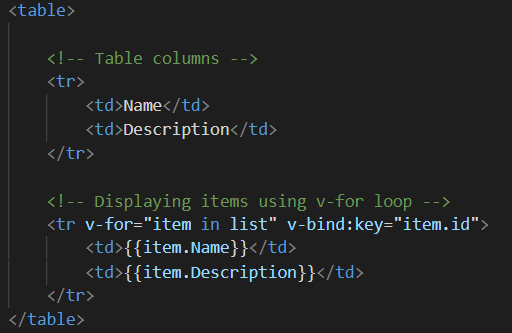
Next, I created the list object.



I then used Axios to retrieve the information from the API response



Similarly, to Angular Vue has a v-for looping call that displays each item as a td element onto the browser.



# Recommendation of framework to use.

Overall, I enjoyed using the Angular framework more than the Vue framework. There are many factors into why I would recommend using Vue over Angular. In my personal experience Angular has a more logical process for frontend web development. I enjoyed using the Angular cli for creating components and services. In the Angular cli you can type ng g c “Component name”. Instantly all the files for the component are created. You can do the same thing for creating services ng g s “Service name”.

I recommend using Angular if you want to create lots of components and services fast. Since the Angular cli already has auto generated files, it is difficult to create errors.

## Pros and Cons of Angular

Angular has been further developed and updates overtime, which provides a steady platform for web development, however, it can take longer to build applications (*Angular vs. Vue – Which Is Best for Programming in 2020?*, 2020).

Using Vuejs can benefit the developer as it takes less time to start prototyping and build an application together. However, Vuejs can be proven to be less advantageous for bigger, extensive applications (*Angular vs. Vue – Which Is Best for Programming in 2020?*, 2020).

A benefit of Vuejs is that it has become widely popular and is being used by more companies everyday (*Angular vs. Vue – Which Is Best for Programming in 2020?*, 2020).

As mentioned before Angular is best for bigger and larger UI applications.

Angular uses two-way data binding to allow for asynchronous services and third-party technology to be incorporated. Whereas, Vue uses one way data flow, it creates smooth data flow which can speed up the development of applications (*Angular vs. Vue – Which Is Best for Programming in 2020?*, 2020).

Angular is known to be more complex and have a harder time to learn and understand official Angular documentation (*Angular vs. Vue – Which Is Best for Programming in 2020?*, 2020). One the other hand Vuejs is easy and simple for first time beginners (*Angular vs. Vue – Which Is Best for Programming in 2020?*, 2020).

In saying this Angular can provide more complexity and flexibility for developing websites. Vuejs does not quite have the same flexibility and sophistication for creating extensive complex applications (*Angular vs. Vue – Which Is Best for Programming in 2020?*, 2020). However, Angular does seem to have a stricter project structure and must follow certain design patters. This can be beneficial as it makes for better coding practices (*Angular vs. Vue – Which Is Best for Programming in 2020?*, 2020).

However, both applications are super-fast and are great solutions for developing web applications.

## My opinion and experiences

Another factor in why I enjoy the Angular framework, over the Vuejs framework is because Angular uses typescript instead of javascript (*Angular vs. Vue – Which Is Best for Programming in 2020?*, 2020). This does create a greater headache in the early process of learning Angular. However, over time typescript enables better and cleaner code then javascript.

Angular elegantly separates the logic from the html. The html is simple and has no extra steps or tags included. The typescript file has a friendly user environment for creating fields, methods and constructors. However, Vuejs has another approach by using a data function to return the values. In Vuejs there is a methods section where all the methods are created. However, doing this way is confusing and less straight forward then having fields and methods created together in a Angular typescript file.

In conclusion I have decided to use Angular, as it makes more sense to me and fits the workflow and style of development, I am comfortable with.

# Table of contents

*Angular vs. Vue – Which is Best for Programming in 2020?* (2020, September 8). FreeCodeCamp.Org. https://www.freecodecamp.org/news/angular-vs-vue-which-is-best-for-programming-in-2020/

Codevolution. (2018, March 6). *Angular Authentication Tutorial—1—Introduction*. https://www.youtube.com/watch?v=ozXGkqpzo\_A&list=PLC3y8-rFHvwg2RBz6UplKTGIXREj9dV0G

*Framework main features—Learn web development | MDN*. (n.d.). Retrieved May 31, 2021, from https://developer.mozilla.org/en-US/docs/Learn/Tools\_and\_testing/Client-side\_JavaScript\_frameworks/Main\_features

*Vue.js JWT Authentication with Vuex and Vue Router—BezKoder*. (n.d.). Retrieved May 31, 2021, from https://bezkoder.com/jwt-vue-vuex-authentication/

Web framework. (2021). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Web\_framework&oldid=1022782175

*What is a web framework? - Evolve evolve*. (n.d.). Retrieved May 30, 2021, from https://evolve.ie/q-and-a/what-is-a-web-framework/