Week 1

Reflection:

This week I have learned HTML, CSS and JavaScript, these three part are the foundation of how to make a websites.HTML is used to build the foundation of the website, CSS is to modify the look of the website, JavaScript is used to let the website have more interactivity of the website.

HTML: can add many element to the website, like other website link, paragraphs or create a table etc.

CSS: modify the look of the website, like background image, font color, font size and style. CSS can be separate to the HTML, one CSS can modify multiple web page in a website.

JavaScript: JavaScript can help website pages to achieve complex functions. It can make the website are not just a simply static information, its real-time content update.

Task1-4

```
39
40      <div id="app">
41          <ul>
42            <li v-for="todo in todos" style="font-size: 20px;">
43              {{  todo.text   }}
44              </li>
45          </ul>
46      </div>
47
48      <script>
49          var app = new Vue({
50              el: '#app',
51              data: {
52                  todos: [
53                      { text: 'Bird' },
54                      { text: 'Horse' },
55                      { text: 'Rhino' }
56                  ]
57              }
58          })
59      </script>
60  </body>
61  </html>
```

```
1   <!DOCTYPE html>
2   <script src="https: //cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
3
4   <style>
5       .title{
6           color: blue;
7           background-color: rgb(8, 18, 158);
8           text-align: center;
9       }
10  </style>
11
12  <html>
13      <script>
14          function myFunction() {
15              document.getElementById("answer").innerHTML = "Rhino";
16          }
17      </script>
18
19  <head>
20      <title>Guess animals</title>
21  </head>
22  <body>
23      <div class= "title">
24          <h1>Guess the animals</h1>
25      </div>
26      <h2 style="text-align: center;"><i>Guess</i></h2>
27      <p style="background-color: orange;color: red; font-size: x-large;"></p>
28
29      <img id="Example" src="1.jpg" alt="Guess the animals" width="400" height="400">
30      <img id="answer" src="2.jpg" alt='Rhino' width="400" height="400" style="display: none;">
31
32      <p>
33          <button type="button" onclick="document.getElementById('Example').src='2.jpg';myFunction()">
34              Click here for answer
35          </button>
36      </p>
37
38      <p id="answer" style="font-size: xx-large;"><b></b></p>
```

Week 2

Reflection:

This week I have learned UX/UI design, is used to structure the project and define entity of the website pages like target audience. It is not easy to do this work, it need to check every position of element and all can successful work as what I designed. I need to consider both practically and aesthetics. When user to use this website, it should not to feel bad about this website, so it was a great challenge for me. Before I started working on the website, I need to think all thing to make sure it will not make a mistake.

Task1-4

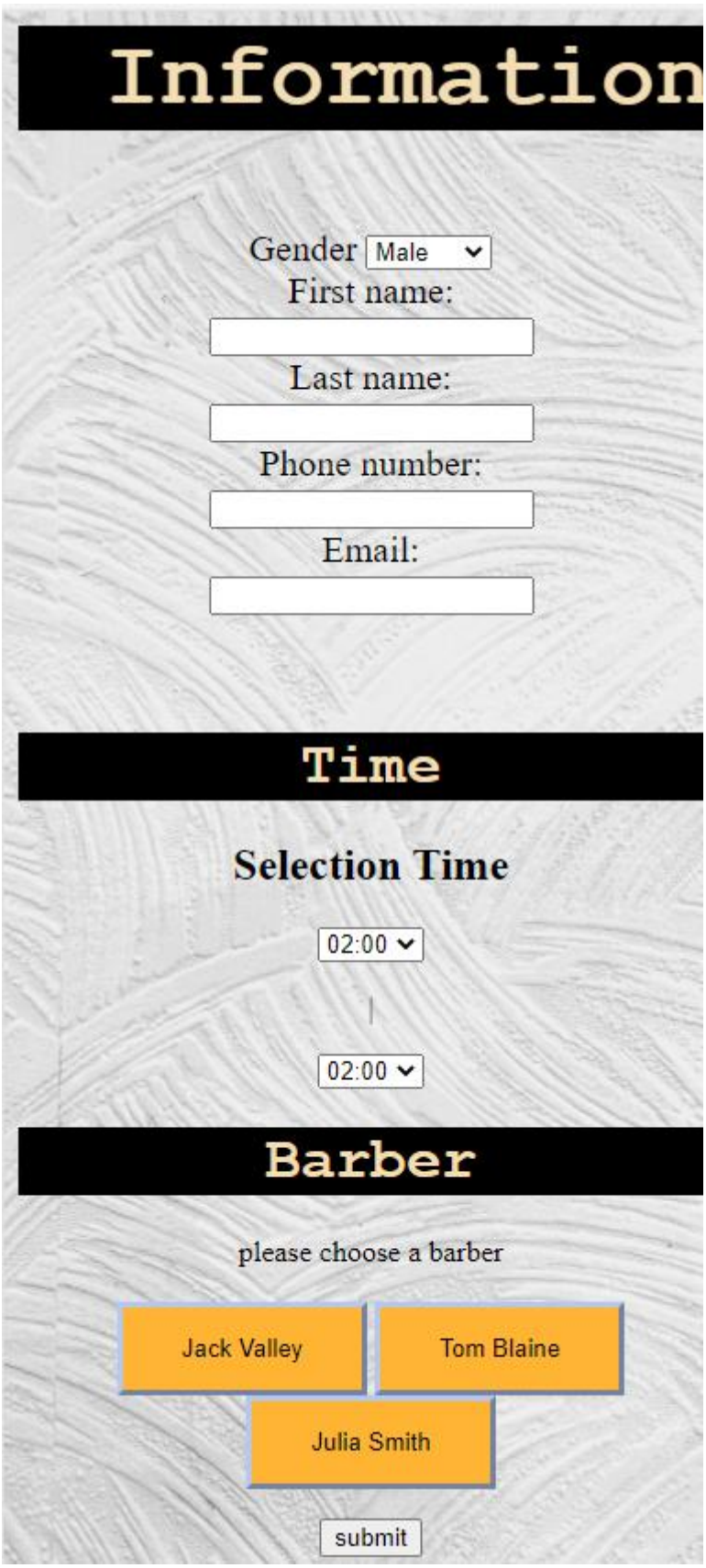
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Test</title>
    <link rel="styleSheet" href="style.css">
</head>
<body>
<h1 style="align-self: start; padding-left: 40px; font-size: 50px;">Information</h1>
    <div class="search_people">
        <div class="search_left">
            <label for="gender">Gender</label>
            <select name="gender" id="gender" class="droplist">
                <option value="Male">Male</option>
                <option value="Female">Female</option>
            </select>
                <form action="/demo/demo_form.asp">
    First name:<br>
    <input type="text" name="firstname" value="">
    <br>
    Last name:<br>
    <input type="text" name="lastname" value="">
    <br>
    Phone number:<br>
    <input type="text" name="Photo number" value="">
    <br>
    Email:<br>
    <input type="text" name="Email" value="">
    <br><br>
    </form>
        </div>
    </div>

    <h1>Time</h1>
    <h2>Selection Time</h2>

<div class="time">
                    <div class="time_set">
                        <select name="time_from" id="start_time" class="time_select">
                                <option value="02:00">02:00</option>
                                <option value="02:30">02:30</option>
                                <option value="03:00">03:00</option>
                                <option value="03:30">03:30</option>
                                <option value="04:00">04:00</option>
                                <option value="04:30">04:30</option>
                                <option value="05:00">05:00</option>
                                <option value="05:30">05:30</option>
                                <option value="06:00">06:00</option>
                                <option value="15:30">15:30</option>
                                <option value="16:00">16:00</option>
                                <option value="16:30">16:30</option>
                                <option value="17:00">17:00</option>
                                <option value="17:30">17:30</option>
                                <option value="18:00">18:00</option>
                                <option value="18:30">18:30</option>
                                <option value="19:00">19:00</option>
                                <option value="19:30">19:30</option>
                                <option value="20:00">20:00</option>
                                <option value="20:30">20:30</option>
                                <option value="21:00">21:00</option>
                                <option value="21:30">21:30</option>
                                <option value="22:00">22:00</option>
                                <option value="22:30">22:30</option>
                                <option value="23:00">23:00</option>
                                <option value="23:30">23:30</option>
                                <option value="24:00">24:00</option>
                                <option value="24:30">24:30</option>
                                <option value="25:00">25:00</option>
                                <option value="25:30">25:30</option>
                        </select>
                        <p id="range_sign">|</p>
                        <select name="time_to" id="end_time" class="time_select">
                                <option value="02:00">02:00</option>
                                <option value="02:30">02:30</option>
                                <option value="03:00">03:00</option>
                                <option value="03:30">03:30</option>
                                <option value="04:00">04:00</option>
                                <option value="04:30">04:30</option>
                                <option value="05:00">05:00</option>
                                <option value="05:30">05:30</option>
                                <option value="06:00">06:00</option>
                                <option value="06:30">06:30</option>
                                <option value="07:00">07:00</option>
                                <option value="07:30">07:30</option>
                                <option value="08:00">08:00</option>
                                <option value="08:30">08:30</option>
                                <option value="09:00">09:00</option>
                                <option value="09:30">09:30</option>
                                <option value="10:00">10:00</option>
                                <option value="10:30">10:30</option>
                                <option value="11:00">11:00</option>
                                <option value="11:30">11:30</option>
                                <option value="12:00">12:00</option>
                                <option value="12:30">12:30</option>
                                <option value="13:00">13:00</option>
```

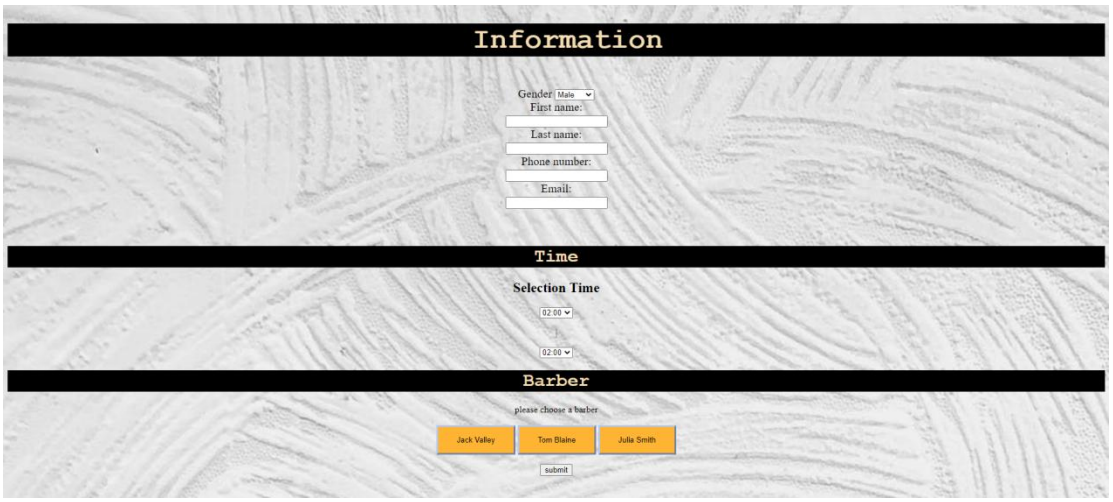
```
                                <option value="15:00">15:00</option>
                                <option value="15:30">15:30</option>
                                <option value="16:00">16:00</option>
                                <option value="16:30">16:30</option>
                                <option value="17:00">17:00</option>
                                <option value="17:30">17:30</option>
                                <option value="18:00">18:00</option>
                                <option value="18:30">18:30</option>
                                <option value="19:00">19:00</option>
                                <option value="19:30">19:30</option>
                                <option value="20:00">20:00</option>
                                <option value="20:30">20:30</option>
                                <option value="21:00">21:00</option>
                                <option value="21:30">21:30</option>
                                <option value="22:00">22:00</option>
                                <option value="22:30">22:30</option>
                                <option value="23:00">23:00</option>
                                <option value="23:30">23:30</option>
                                <option value="24:00">24:00</option>
                                <option value="24:30">24:30</option>
                                <option value="25:00">25:00</option>
                                <option value="25:30">25:30</option>
                            </select>
                        </div>
    </p>

    <h1>Barber</h1>
    <p>please choose a barber<br><br>
    <button class="button" type="button">Jack Valley </button>
    <button class="button" type="button">Tom Blaine</button>
    <button class="button" type="button">Julia Smith</button>
    </p>

    <div class="submit">
        <a href="end.html">
        <button type="button">submit</button>
    </div>




</body>
</html>
```

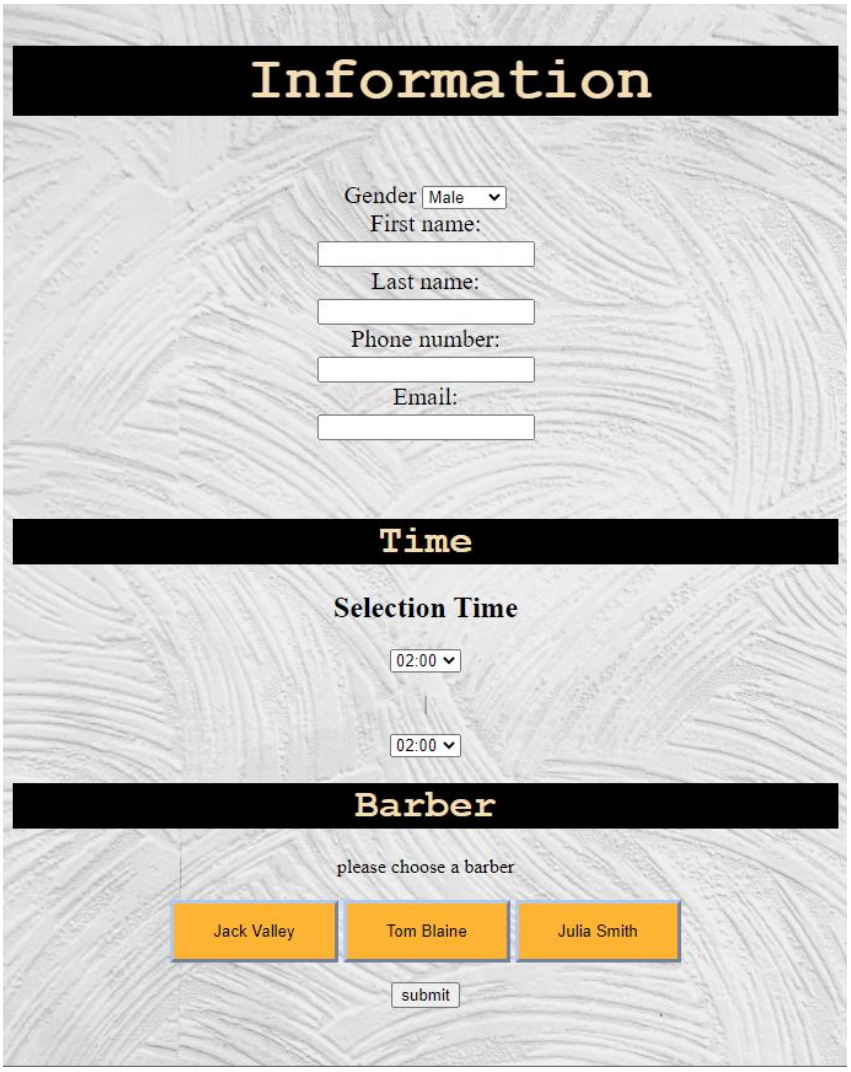
CSS file

```css
h1{
    background-color: black;
    font-family: 'Courier New', Courier, monospace;
    color: wheat;
    text-align: center;
}
h2{
    text-align: center;
}
p{
    font: 15px 'courier new' bold;
    text-align: center;

}
.time{
    text-align: center;
}


.button:hover{
    background-color: #b6d11f;
    color: black;
}

.button{
    border:3px outset rgb(179, 200, 245);
    color: black;
    background-color:#FFB533;


    padding: 15px 32px;
    text-align: center;

}
.login{
    text-align: center;
}

.registered{
    text-align: center;
}
.end{
    text-align: center;
}
.search_left{
    text-align: center;
    font-size: 20px;
    padding: 20px 40px;
}
```

```css
.search_left{
    text-align: center;
    font-size: 20px;
    padding: 20px 40px;
}
.david1{
    border:wheat;
    border-radius: 30%;
}
img{
    background-color: wheat;
}
body{
    background-image:url("https://images.pexels.com/photos/1939485/pexels-photo-1939485.jpeg?auto=compress&cs=tinysrgb&dpr=2&h=750&w=1260") ;
    background-repeat: no-repeat;
    background-position: top center;
    background-size:cover ;
}
```


Information
Gender Male
First name:
Last name:
Phone number:
Email:
Time
Selection Time
02:00
02:00
Barber
please choose a barber
Jack Valley
Tom Blaine
Julia Smith
submit





User story:

1. As a user, I want a haircut and I'm not sure if the haircut time conflict

with my schedule, I can through the website to make an appointment and prevent time conflict. Users can see the timetable in booking page and to check the time.

2. As a user, I'm a regular customer of this barber shop and I know whichbarber is the best, when I making an appointment, I can choose the

barber who I want.

3. As a user, I want to go to the barber shop near me, before you start make a appointment,

you can choose a barber shop, and check which

one is close-in.

4. As a user, I want to make a appointment, but I don't want to register my information every

time, so the user can

create a account, this can

save a lot of time for users.

5. User process website to appointment,

they will need not have

communicate with receptionist.

6. As a mobile user, I can process this website to make a appointment at any time. So our website should be a responsive, there is another stylling made for a small screen

Week 3

Reflection:

In this week, I have learn more, and I can improve my web page, let the users can have more good experience when they use and visited website pages. And I have add more function can give user more good experience. It is not just a simple website page for the users, I had searched some example and repeat checking the web page for improving the gallery. And I have add a to top button to help the users, this function can be apply to every kind of web page.

Tasks

```
<head>
    <meta charset="UTF">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Test</title>
    <link rel="styleSheet" href="a.css">
</head>
<body>
<h1 style="align-self: start; padding-left: 40px; font-size: 50px;">Login</h1>
    <div class="search_people">
        <div class="search_left">

                    <form action="/demo/demo_form.asp">
    Username:<br>
    <input type="text" name="username" value="">
    <br>
    Password:<br>
    <input type="text" name="password" value="">
    <br>
    </form>
            </div>
        </div>
        <div class="login">
            <a href="index.html">
            <button type="button">login</button>
        </div>
        <div class="registered">
            <a href="registered.html">
            <button type="button">registered</button>
        </div>

    </body>
    </html>
```

```html
<head>
    <meta charset="UTF">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Test</title>
    <link rel="styleSheet" href="a.css">
</head>
<body>
<h1 style="align-self: start; padding-left: 40px; font-size: 50px;">Registered</h1>
    <div class="search_people">
        <div class="search_left">

                    <form action="/demo/demo_form.asp">
    Username:<br>
    <input type="text" name="username" value="">
    <br>
    Password:<br>
    <input type="text" name="password" value="">
    <br>
    Confirm Password:<br>
    <input type="text" name="Epassword" value="">
    <br>
    Phone number:<br>
    <input type="text" name="Phone" value="">
    <br>
    <div class="Varcode">
        <input type="text" value="" placeholder="Enter the Varcode" class="input-val">
        <canvas id="canvas" width="100" height="30"></canvas>
        <button class="btn">submite</button>
    </div>

    <div class="login">
        <a href="index.html">
        <button type="button">submit</button>
    </div>
    <div class="back">
        <a href="login.html">
        <button type="button">back</button>
    </div>


</body>
```

```css
#toTop {
    display: none;
    position: fixed;
    bottom: 20px;
    right: 30px;
    z-index: 99;
    border: none;
    outline: none;
    background-color:  #55c3ad;
    color: white;
    cursor: pointer;
    padding: 15px;
    border-radius: 10px;
    font-size: 18px;
  }
```

```css
#toTop:hover {
    background-color: #42bda4;
}
```

Week 4

Reflection:

In this week, I have learned vue Frameworks. Frameworks is designed for reduce the web develop works. Vue is a client side frameworks, it can make the website become interactive. When we want to develop the website, we will use the HTML, CSS and JavaScript to build this website, HTML used to created the appearance of the website, CSS used to modify the look of the website and JavaScript it to show the behaviour of the website, to make a interaction with users. The frameworks is like we have a phone, and it have install all application what you need, so you can easily to use this phone.

Tasks:

```html
<!DOCTYPE html>
<link href="stylesheet.css" rel="stylesheet">
<link href="script.js" rel="stylesheet">

<html>

<style>
    a{
        text-decoration: underline;
        min-width: 250px;
    }
</style>
<title>
    Login
</title>

<body>
    <form id="login">
        <h2>sign in</h2><br>
        <input id="Email" type="email" placeholder="Enter your email"><br>
        <input id="PW" type="text" placeholder="password"><br>
        <a id="SwitchSignUp">Create a account</a><br>
        <button id="LoginBtn" type="submit">Login</button>
    </form>

    <form id="SignUp">
        <h2>Creat your Account</h2>
        <input id="FirstName" type="text" placeholder="First Name"><br>
        <input id="LastName" type="text" placeholder="Last Name"><br>
        <input id="Email" type="email" placeholder="Email"><br>
        <input id="DOB" type="text" placeholder="Date of Birthday"><br>
        <input id="Password" type="text" placeholder="Password"><br>
        <a id="SwitchSignUp">Login</a><br>
        <button id="submitbtn" type="submit">Submit</button>
    </form>
</body>

</html>
```

```
var LoginForm = document.getElementById("login");
var SwitchToSignUp = document.getElementById("SwitchSignUp");
var LoginBtn = document.getElementById("LoginBtn");

SwitchToSignUp.onclick = function () {
    LoginBtn.style.display = "none";
    SignUpForm.style.display = "block";
}

var SignUpForm = document.getElementById("SignUp");
var SwitchToLogin = document.getElementById("SwitchLogin");
var submitbtn = document.getElementById("submitbtn");

SignUpForm.style.display = "none";
SwitchToLogin.onclick = function () {
    SignUpForm.style.display = "none";
    LoginForm.style.display = "block";
}

var FirstName = document.getElementById("FirstName").value;
var LastName = document.getElementById("LastName").value;
var Email = document.getElementById("Email").value.indexOf("@");
var DOB = document.getElementById("DOB").value;
var Password = document.getElementById("Password").value;

LoginBtn.onclick = function () {
    submit = "true";
    if (Email == -1) {
        window.alert("Please enter a email");
        submit = "fales";
    }
    if (Password.Length > -1 && Password.Length < 4) {
        window.alert("The password at least need have 5 characters");
        submit = "false";
    }
    if (submit == "false") {
        return false;
    }
}
```

```css
#login{
    line-height: 50px;
    margin-right: 40%;
    margin-left: 40%;
    max-width: 250px;
}

#login input{
    width: 250px;
    height: 30px;
}

#LoginBtn{
    width: 50px;
    height: 30px;
}

#SignUp{
    line-height: 50px;
    margin-left: 40%;
    margin-right: 40%;
    max-width: 250px;
}

#SignUp input{
    width: 250px;
    height: 30px;
    background-color: white;
}

#SignUp :hover{
    background-color: gray;
}

#submitbtn{
    width: 60px;
    height: 30px;
}
```

sign in

Enter your email

password

Create a account

Login

Creat your Account

First Name

Last Name

Email

Date of Birthday

Password

Login

Submit

Week 5

Reflection:

In this week, I have started to learned Vue.js framework, like loops and user input. We can put the sentence into Vue, the sentence can be show on the website pages. And I have learned how to enter and output the value on the website pages. This is helpful for make a website. We can process Vue framework to improve the todo list, we can to add or delete todo text. And we can set what we want in to Vue.component, the component can put into script.

Task

Understand components in Vue:

Components are blocks of code, when save it in a script sheet, it can be reuse in any time, just need to add the components name with<> in the HTML file, and it will be show on the website page.

```html
<!DOCTYPE html>

<html>

<head>
    <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
</head>

<body>
    <h3>Task 1</h3>
    <div id="app">
        <p>{{ message }}</p>
        <p>{{ detail() }}</p>

        <p v-html="task2"></p>
        <p v-html="img"></p>
    </div>

    <h3>This is task 3:</h3>
    <div id="app2">
        <span v-if="seen">Test text</span><br>
        <h3>Task4</h3>
        <p>add something to List</p>
        Enter todo:<input ref="update">
        <button v-on:click="update">add</button>
        <ul>
            <li v-for="todo in todos">
                {{  todo.text   }}
            </li>
        </ul>
    </div>

    <script>
        var app = new Vue
        ({
            el: '#app',
            data: {
                message: 'hello world',
                name: 'Jacky',
                address: 'Burwood',
                task2: '<h3>Task2</3>',
                img: '<img src=3.jpg width=200px height=250px>'
            },
            methods: {
                detail: function() {
                    return "My name is " + this.name + "and now I live in" + this.address;
                }
            }
        })
```

```
        var app2 = new Vue({
            el: '#app2',
            data: {
                seen:true,
                todos: [
                    { text: 'January' },
                    { text: 'February'},
                    { text: 'March'},
                ]
            },

            methods: {
                update: function(){
                    this.todos.push({ text: this.$refs.update.value })
                }
            }
        })

        var app3 = new Vue({
            el: '#component'
        })
    </script>

</body>
</html>
```

Task 1

hello world

My name is Jackyand now I live inBurwood

Task2

This is task 3:

Test text

Task4

add something to List

Enter todo: add

- January
- February
- March

Week 6

Reflection:

In this week I have learned how can make a dynamic component by Vue and the different between global and local component. Component are re-usable instance of code that can be stacked up in a tree to effectively break down code into manageable sections to work on avoid have to repetitively edit large areas of code. The computed methods can help to output the result easier. And the watch methods can help to convert data, like currency converter, enter the AUD convert to USD.

Task

```
<html>

<head>
    <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
</head>

<body>
    <h1> Task 1</h1>
    <div id="component1">
        app1:
    <global-component></global-component>

    <local-component></local-component>
    </div>

    <div id="component2">
        app2:
        <local-component></local-component>
    </div>

    <div id="dynamic_binding">
        app3:
        <component v-bind:is="dynamic_entry"></component>
        <dynamic-component></dynamic-component>
    </div>

    <script>
        Vue.component('global-component', {
            template: '<h3> Here is a global_component</h3>'
        })
        var app1 = new Vue({
            el: '#component1',
            data:{

            }
        })

        var app2 = new Vue({
            el: '#component2',

            components: {
                'local-component':{
                    template: '<h3> Local_component</h3>'
                }
            }
        })
```

```
        Vue.component('dynamic-component',{
            template:'<h3>Component passed as data porperty to app3</h3>'
        })

        var app3 = new Vue({
            el: '#dynamic_binding',
            data: {
                dynamic_entry: 'dynamic-component'
            }
        })
    </script>

</body>
```

Task 1

app1:

Here is a global_component

app2:

Local_component

app3:

Component passed as data porperty to app3

Component passed as data porperty to app3

```
!DOCTYPE html>

<html>

<head>
    <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
</head>

<body>
    <h1>Task 2</h1>

    <div id="component1">
        <component_1></component_1>
    </div>

    <script>
        Vue.components('component_1', {
            template:'<div v-on:mouseover="changename()"><h3><span id = name>{{ name }}</spam></h3></div>',
                data: function(){
                    return{
                        name: "hello world"
                    }
                },

                methods: {
                    changename : function(){
                        this.name = "Thanks"
                    }
                }
        })
        var app1 = new Vue({
            el:'$component1',
            data:{

            }
        })
    </script>
```

```html
<!DOCTYPE html>

<html>
    <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>

    <body>
        <h1>Task3</h1>
        <div id='app1'>
            First Name: <input type="text" v-model = "firstname">
            Last Name:<input type="text" v-model = "lastname"><br>

            <p>{{ firstname}} {{lastname}}</p>
            computed name: <h3>{{ getname}}</h3>
        </div>

        <div id="app2">
            Reverse a message:<input type="text" v-model = "message"><br>
            reversed message:<h3>{{reversedMessage}}</h3>
        </div>

        <script>
            var app1 = new Vue({
                el:'#app1',
                data:{
                    firstname:'',
                    lastname:'',
                },
                computed:{
                    getname: function(){
                        return this.firstname + "" + this.lastname
                    }
                }
            })

            var app2 = new Vue({
                el:'#app2',
                data:{
                    message:''
                },
                computed:{
                    reversedMessage: function(){
                        return this.message.split('').reverse().join('')
                    }
                }
            })
        </script>
    </body>

</html>
```

Task3

First Name: Jack Last Name: Smith

Jack Smith

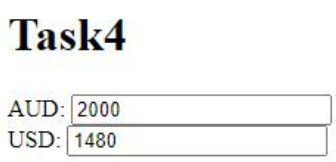
computed name:

JackSmith

Reverse a message: pizza
reversed message:

azzip

```
<!DOCTYPE html>

<html>
    <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>

    <body>
        <h1>Task4</h1>

        <div id="app1">
            AUD: <input type="text" v-model = "AUD"><br>
            USD: <input type="text" v-model = "USD"><br>
        </div>

        <script>
            var app1 = new Vue({
                el:'#app1',
                data:{
                    AUD:'0',
                    USD:'0'
                },
                watch:{
                    AUD:function(val){
                        this.AUD = val;
                        this.USD = val * 0.74;
                    },
                    USD:function(val){
                        this.USD = val;
                        this.AUD = val / 0.74;
                    }
                }
            })

        </script>

    </body>

</html>
```



Week 7

Reflection:

In this week I have learned how to use props, transition, slots and animation. And leaned how to use dynamic and Async components in the website pages. There are some new dynamic component, it can improve the experience of the use website pages. It can be used in many project. It allow us to learned some more complex elements of working with Vue components. Process these new elements like transitions, animations and props can be great help and improvement for me to revise and refine my project.

Task1

Props can be through the component and set the type of the element. Props are used to pass data down from a parent component to a child.

```html
<!DOCTYPE html>

<html>
    <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>

    <body>
        <div id='app1'>
            <component1 v-bind:content="datacontent"></component1>
        </div>
    </body>

    <script>
        Vue.component('component1',{
            props:['content'],
            template:'<h3>{{content}}</h3>'
        })
        new Vue({
            el:'#app1',
            data:{
                datacontent:'this data is rednerd Props'
            }
        })
    </script>
</html>
```

Task2:

Slots allow me to using another component, another component is mean what we have created. Slots let component combine with other outside of the parent-child tree.

```
<!DOCTYPE html>

<html>
    <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
    <body>

        <div id="app1">
            <slots v-slot:prop>
                <component1></component1>
            </slots>

        </div>
    </body>
    <script>
        Vue.component('slots',{
            props:['prop'],
            template:'<div><h3><slot name="prop"></slot></h3></div>',
        })

        Vue.component('component1',{
            template:'<div><p><slot>text in component </slot></p></div>'
        })
        new Vue({
            el:'#app1',
            data:{
                label:"slots"
            }
        })
    </script>

</html>
```

Task3

I have learned <keep-alive>, it is to help s can return to the last statement and then we go to the next page, it will not to show the post what we select.

Task4

```
<!DOCTYPE html>

<html>
    <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
    <body>
        <style>
            .fade-enter-active,
            .fade-enter-active{
                transition: opacity 2.5s;
            }

            .fade-enter,
            .fede-leave-to
            {
                opacity: 0;
            }

            .bounce-enter-active{
                animation: bounce-in 5s;
            }

            .bounce-enter-active{
                animation: bounce-in 5s reverse;
            }

            @keyframes bounce-in{
                0%{
                    transform:scale(0);
                }
                50%{
                    transform:scale(1.5);
                }
                100%{
                    transform: scale(1);
                }
            }
        </style>
        <div id="app">
            <button v-on:click="show = !show">Click</button>
            <transition name="fade">
                <p v-show="show"><img src="3.jpg" style="width: 150px; height: 200px;"></p>
            </transition>
        </div>
        <script>
            var app = new Vue({
                el:"#app1",
                data:{
                    show: true
                }
```

```
49                  }
50              });
51              var app1 = new Vue({
52                  el:'#app2',
53                  data:{
54                      show:true
55                  }
56              })
57          </script>
58      </body>
59  </html>
```