

8. 會員系統(註冊)

1. 註冊畫面

會員註冊

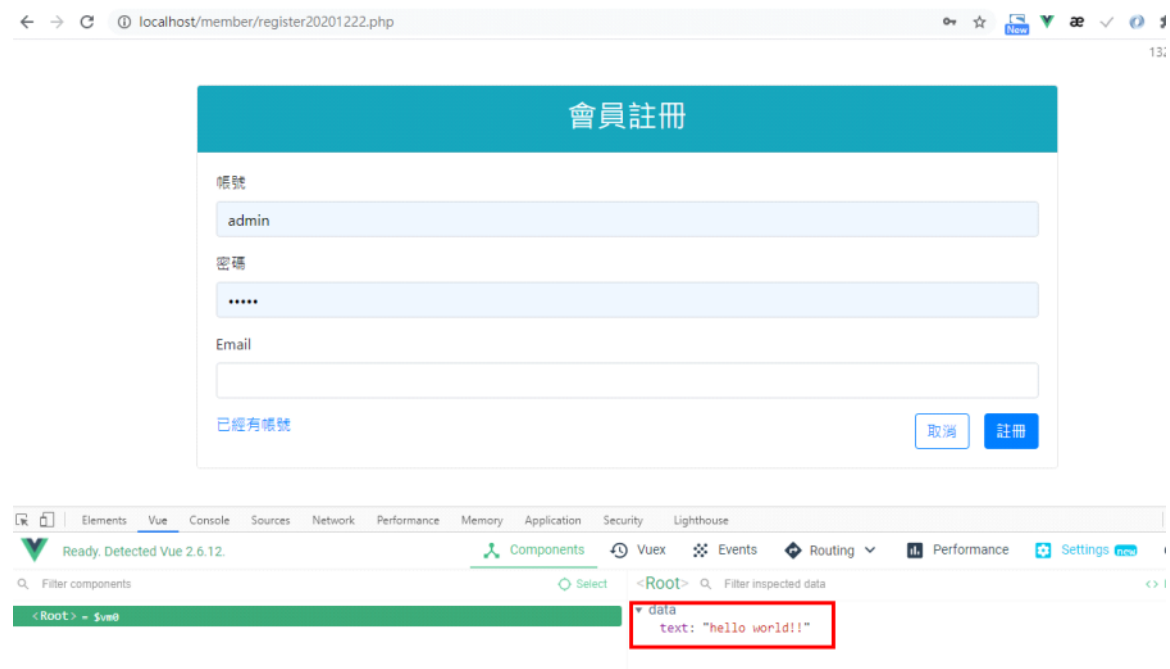
帳號

密碼

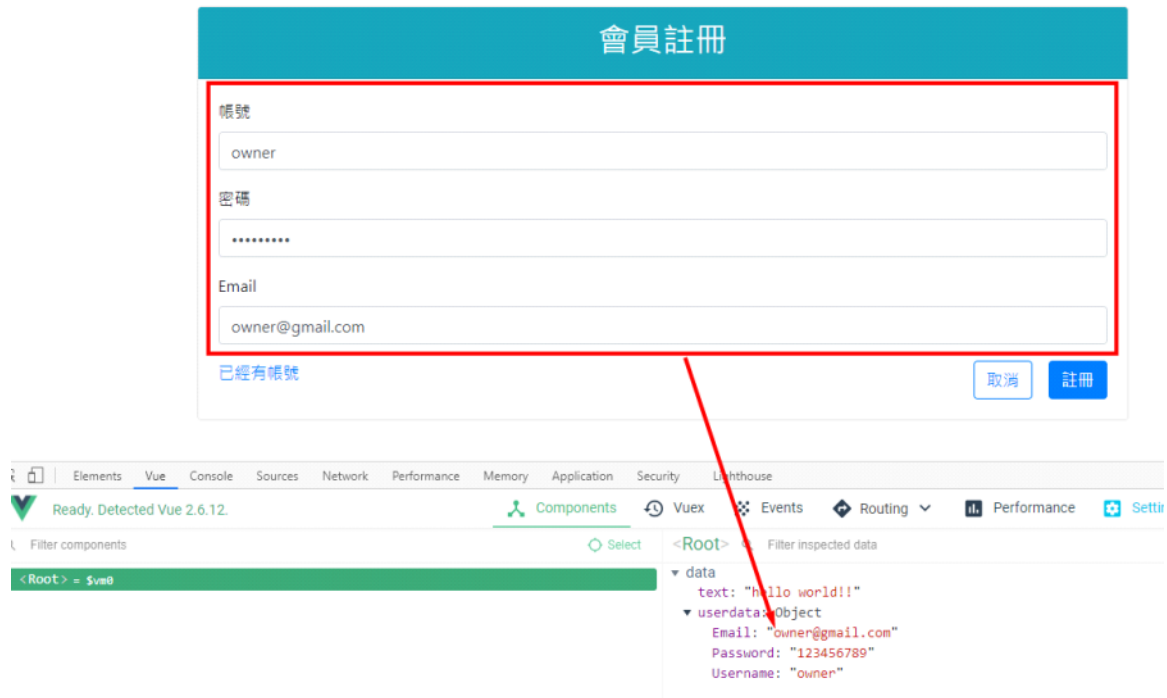
Email

[已經有帳號](#)

使用vue功能



綁定資料



使用axios 將資料傳遞到api/register-api.php

需使用FormData()為傳遞的資料型態

```
data: {
  text: 'hello world!!',
  Username: '',
  UsernameError: false,
  UsernameErrmsg: '',
  Password: '',
  PasswordError: false,
  PasswordErrmsg: '',
  Email: '',
  EmailError: false,
  EmailErrmsg: ''
},
```

```
methods: {
  register: function(){
    vm = this; //避免重複使用this
    //FormData 介面可為表單資料中的欄位/值建立相對應的的鍵/值對 (key/value) 集合
    var mydata = new FormData();
    mydata.append('username', vm.Username);
    mydata.append('password', vm.Password);
    mydata.append('email', vm.Email);
    // Send a POST request
    axios({
      method: 'post',
      url: 'api/register-api.php',
      data: mydata
    }).then(function (response) {
      console.log(response);
    }).catch(function (error) {
      console.log(error);
    });
  }
};
```

```

    }).catch(function (error) {
      console.log(error);
    });
  },
},

```

The screenshot displays a web application with a teal header titled "會員註冊" (Member Registration). The form contains three input fields: "帳號" (Username) with the value "owner001", "密碼" (Password) with masked characters "*****", and "Email" with the value "owner001@gmail.com". Below the inputs are two buttons: "取消" (Cancel) and "註冊" (Register). A link "已經有帳號" (Already have an account) is also present.

The Vue DevTools console shows the following log messages:

```

You are running Vue in development mode.
Make sure to turn on production mode when deploying for production.
See more tips at https://vuejs.org/guide/deployment.html

vue-devtools Detected Vue v2.6.12

{data: {…}, status: 200, statusText: "OK", headers: {…}, config: {…}, …}
register2

{data: {…}, status: 200, statusText: "OK", headers: {…}, config: {…}, …}
register2

{data: {…}, status: 200, statusText: "OK", headers: {…}, config: {…}, …}
register2

{data: {…}, status: 200, statusText: "OK", headers: {…}, config: {…}, …}
register2

```

The expanded log message shows the following data:

```

data: {
  message: "insert ok!"
  state: true
  __proto__: Object
}
headers: {connection: "keep-alive", content-length: "90", content-type: "text/html; charset=UTF-8", date: "Tue, 22 Dec 2020 07:18:42 GMT", keep-alive: "timeout=5, max=100", …}
request: XMLHttpRequest {readyState: 4, timeout: 0, withCredentials: false, upload: XMLHttpRequestUpload, onreadystatechange: f, …}
status: 200
statusText: "OK"
__proto__: Object

```

2. 使用正規表達式訂定義輸入規則

帳號: 當不符合正規表達式或字數規定時更改UsernameError和UsernameErrmsg

```

watch: {
  Username: function(){
    vm = this; //避免重複使用this
    var isText = /^[a-zA-Z0-9]+$/;
    // test() 方法用於檢測一個字符串是否匹配某個模式
    if (!isText.test(vm.Username)) {
      vm.UsernameError = true;
      vm.UsernameErrmsg = '請勿包含特殊字元';
    } else if (vm.Username.length > 10) {
      vm.UsernameError = true;
      vm.UsernameErrmsg = '請勿超過10個字';
    } else {
      vm.UsernameError = false;
    }
  }
}

```

is-invalid 搭配 invalid-feedback (bootstrap內建)

```

<div class="form-group">
  <label for="username">帳號</label>
  <input type="text" id="username" class="form-control"
    :class="{ 'is-invalid' : UsernameError}" v-model="Username">
  <div class="invalid-feedback">
    {{ UsernameErrmsg }}
  </div>
</div>

```

不符合 正規表達式 /^[a-zA-Z0-9]+\$/

會員註冊

帳號

owner@
❗

請勿包含特殊字元

密碼

Email

已經有帳號

取消
註冊

不符合字數

會員註冊

帳號

❗

請勿超過10個字

密碼

Email

[已經有帳號](#) [取消](#) [註冊](#)

密碼與Email

```
Password: function(){
  vm = this; //避免重複使用this
  var isText = /^[a-zA-Z0-9]+$//; //包含26字母和0-9
  var include = /^(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,15}$/; //包含0-9, a-z, A-Z 各一個

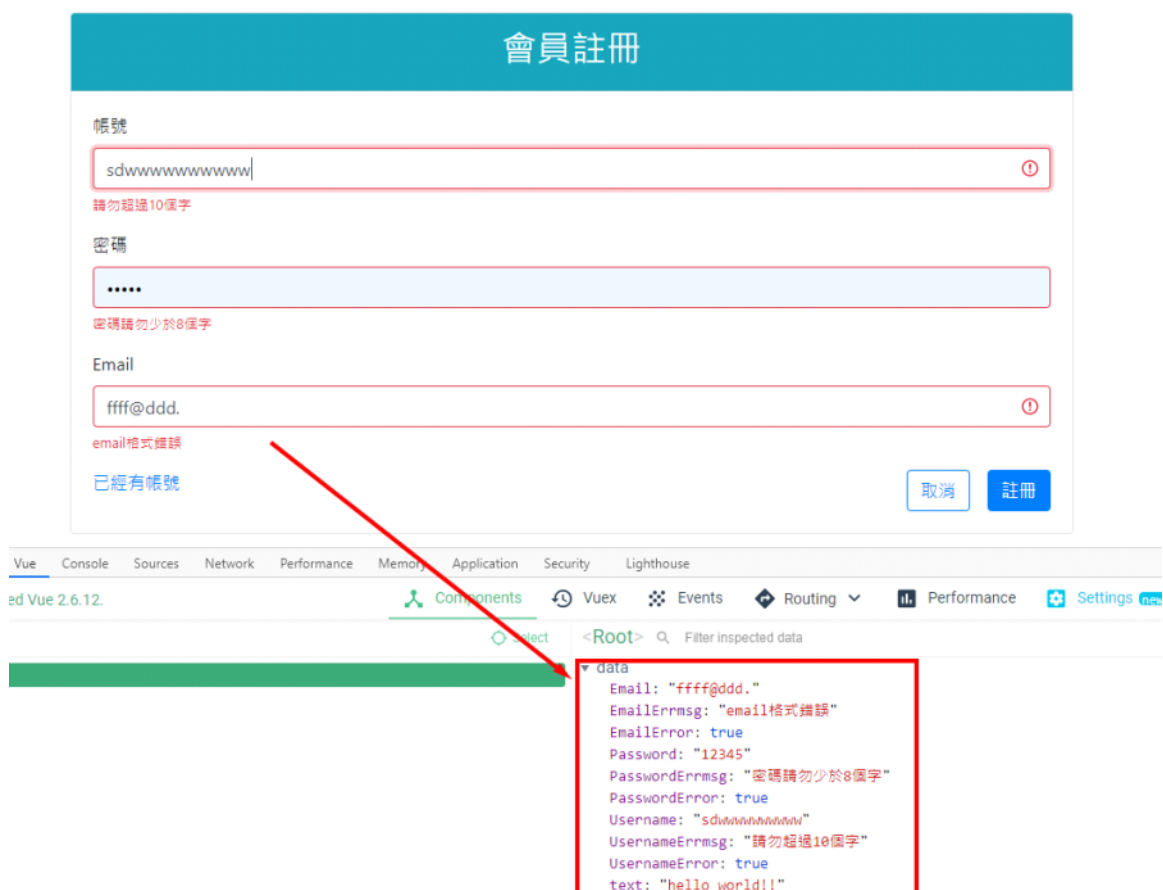
  if(!isText.test(vm.Password)){
    vm.PasswordError = true;
    vm.PasswordErrmsg = '請勿包含特殊字元';
  }else if (vm.Password.length < 8){
    vm.PasswordError = true;
    vm.PasswordErrmsg = '密碼請勿少於8個字';
  }else if (vm.Password.length > 15) {
    vm.PasswordError = true;
    vm.PasswordErrmsg = '請勿超過15個字';
  }else if (!include.test(vm.Password)) {
    vm.PasswordError = true;
    vm.PasswordErrmsg = '至少包括一個大小寫字母或數字';
  }else{
    vm.PasswordError = false;
  }
},
```

```
Email: function(){
  vm = this; //避免重複使用this
  var isMail = /^\\w+((-\\w+)|(\\.\\w+))*\\@[A-Za-z0-9]+((\\.|-)\\[A-Za-z0-9]+)*\\.\\[A-Za-z]+$//;
  if (!isMail.test(vm.Email)) {
    vm.EmailError = true;
    vm.EmailErrmsg = 'email格式錯誤';
  }
  else {
    vm.EmailError = false;
  }
}
```

```

<div class="card-body">
  <div class="form-group">
    <label for="username">帳號</label>
    <input type="text" id="username" class="form-control"
      :class="{ 'is-invalid' : UsernameError}" v-model="Username">
    <div class="invalid-feedback">
      {{ UsernameErrmsg }}
    </div>
  </div>
  <div class="form-group">
    <label for="password">密碼</label>
    <input type="password" id="password" class="form-control"
      :class="{ 'is-invalid': PasswordError}" v-model="Password">
    <div class="invalid-feedback">
      {{ PasswordErrmsg }}
    </div>
  </div>
  <div class="form-group">
    <label for="email">Email</label>
    <input type="email" id="email" class="form-control"
      :class="{ 'is-invalid': EmailError}" v-model="Email">
    <div class="invalid-feedback">
      {{ EmailErrmsg }}
    </div>
  </div>
  <div class="d-flex">
    <a href="login.php">已經有帳號</a>
    <button class="btn btn-outline-primary ml-auto">取消</button>
    <button class="btn btn-primary ml-3" @click="register">註冊</button>
  </div>
</div>

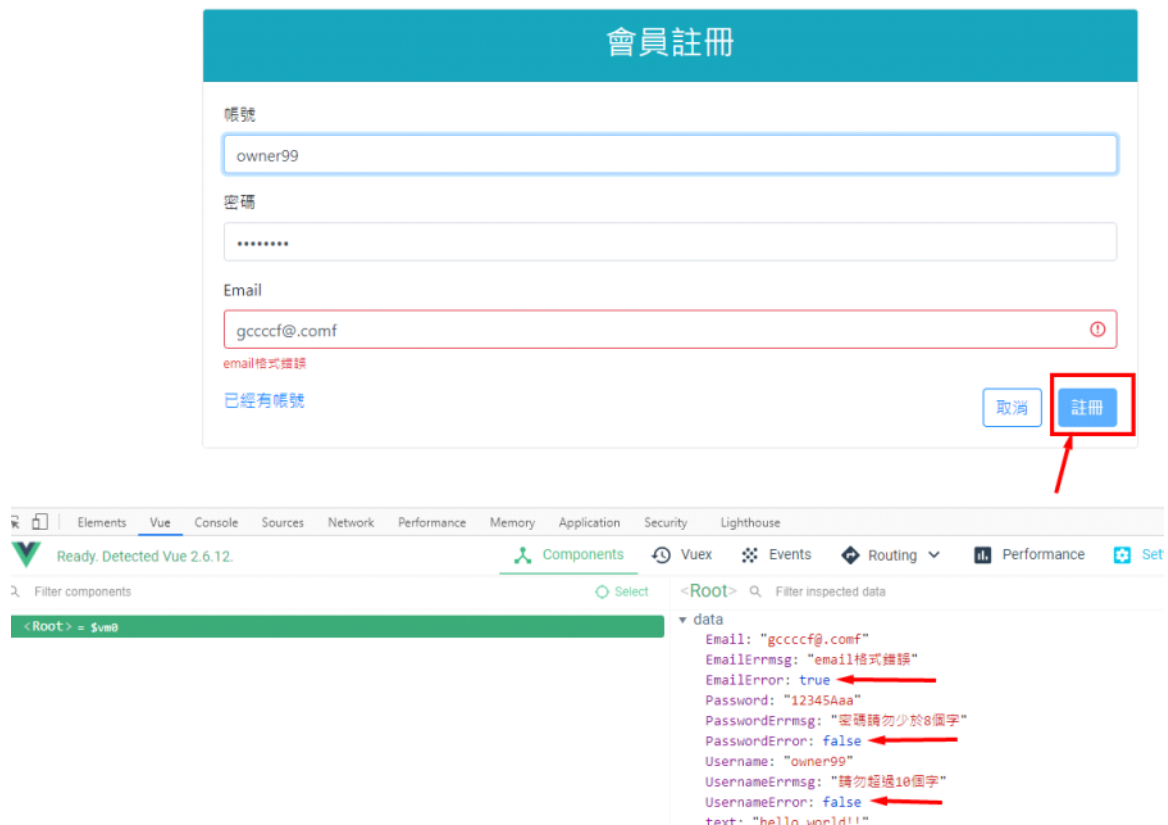
```



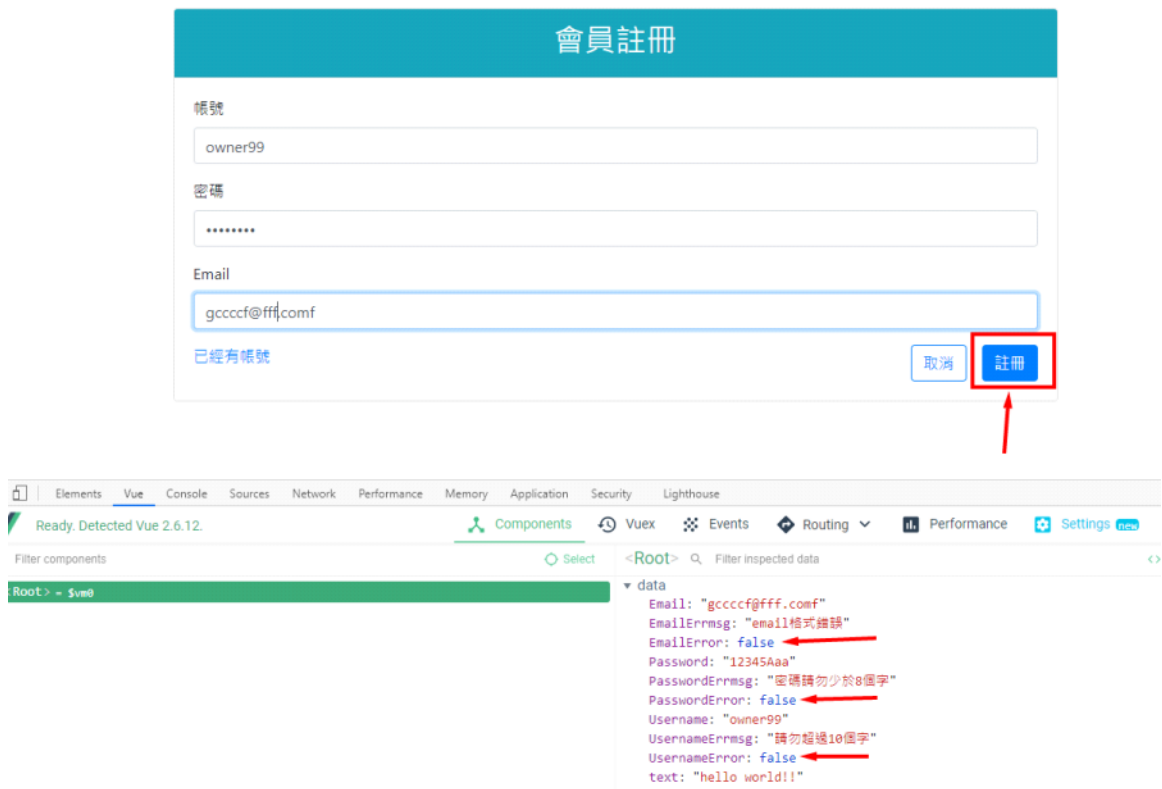
3. 設定兩個按鈕並使用v-if v-else 判斷三個欄位都符合規定時，正確的按鈕才能被使用，也才能傳遞資料給API!

```
<div class="d-flex">
  <a href="login.php" class="mt-3">已經有帳號</a>
  <button class="btn btn-outline-primary ml-auto">取消</button>
  <button class="btn btn-primary ml-3 disabled"
    v-if="UsernameError||PasswordError||EmailError">註冊</button>
  <button class="btn btn-primary ml-3" v-else @click="register">註冊</button>
</div>
```

只要有一個不符合規定按鈕就不能使用!



全部符合規定按鈕才能使用!



4. 判斷帳號是否已存在

判斷帳號的前提為帳號必須符合規則!

```
Username: function(){
  vm = this; //避免重複使用this
  var isText = /^[a-zA-Z0-9]+$/;
  // test() 方法用於檢測一個字符串是否匹配某個模式
  if (!isText.test(vm.Username)) {
    vm.UsernameError = true;
    vm.UsernameErrMsg = '請勿包含特殊字元';
  } else if (vm.Username.length > 10) {
    vm.UsernameError = true;
    vm.UsernameErrMsg = '請勿超過10個字';
  } else {
    vm.UsernameError = false;
  }

  //符合輸入規則時才判斷是否有帳號存在
  if(!vm.UsernameError){
    // check_uni
    var mydata = new FormData();
    mydata.append('username', vm.Username);
    axios({
      method: 'post',
      url: 'api/register-check-uni-api.php',
      data: mydata
    }).then(function (response) {
      console.log(response);
      // (response.data.state true:存在, 不可以使用 false: 不存在, 可以使用)
      vm.UsernameError = response.data.state;
      vm.UsernameErrMsg = response.data.message;
    }).catch(function (error) {
      console.log(error);
    });
  }
}
```


利用invalid-feedback與valid-feedback搭配class綁定is-invalid與is-valid顯示提示訊息!

```
<div class="form-group">
  <label for="username">帳號</label>
  <input type="text" id="username" class="form-control"
    :class="{ 'is-invalid' : UsernameError, 'is-valid' : !UsernameError }"
    v-model="Username">
  <div class="invalid-feedback">
    {{ UsernameErrMsg }}
  </div>
  <div class="valid-feedback">
    {{ UsernameErrMsg }}
  </div>
</div>
```

搭配Vue.js devtools觀察其數值變化!

會員註冊

帳號

owner999

帳號不存在,可以使用!

密碼

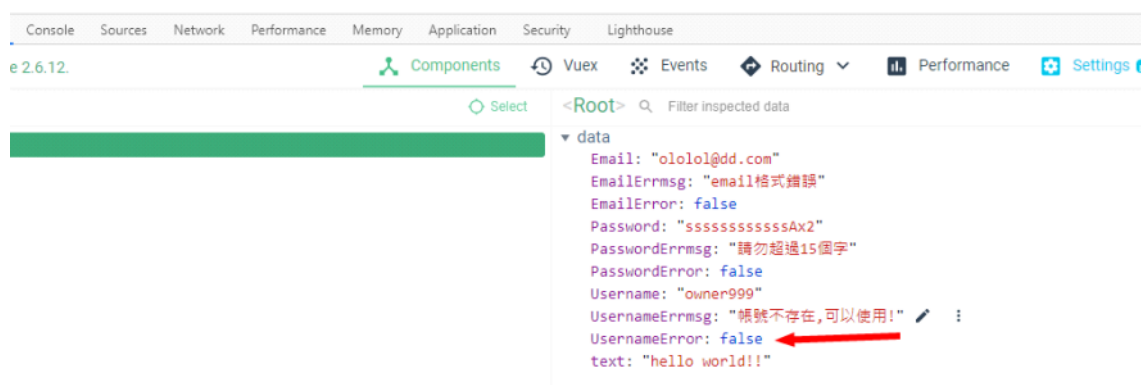
Email

ololol@dd.com

已經有帳號

取消

註冊



會員註冊

帳號

 ⓘ
帳號已存在,不可使用!

密碼

 ⓘ
密碼不得為空白

Email

 ⓘ
Email不得為空白

[已經有帳號](#) [取消](#) [註冊](#)

會員註冊

帳號

 ⓘ
請勿超過10個字

密碼

 ⓘ
密碼不得為空白

Email

 ⓘ
Email不得為空白

[已經有帳號](#) [取消](#) [註冊](#)

5. 註冊成功後跳轉至登入頁面

```

methods: {
  register: function(){
    vm = this; //避免重複使用this
    //FormData 介面可為表單資料中的欄位/值建立相對應的的鍵/值對 (key/value) 集合
    var mydata = new FormData();
    mydata.append('username', vm.Username);
    mydata.append('password', vm.Password);
    mydata.append('email', vm.Email);
    // Send a POST request
    axios({
      method: 'post',
      url: 'api/register-api.php',
      data: mydata
    }).then(function (response) {
      console.log(response);
      if(response.data.state){
        alert(response.data.message);
        location.href = "login20201223.php";
      }
    }).catch(function (error) {
      console.log(error);
    });
  },
},

```

