

<p style="text-align: center;">ปฏิบัติการที่ 4</p> <p style="text-align: center;">วิชา 523419 Advanced Web Application Development</p> <p>รหัสนักศึกษา..... ชื่อ.....</p> <p><u>วัตถุประสงค์</u></p> <p><u>ไฟล์ที่จำเป็น</u></p>	<p>คะแนน</p>
--	---------------------

แบบฝึกปฏิบัติการที่ 4.1 Reactive Form: Form Control

1. สร้างแอปพลิเคชันใหม่ด้วยคำสั่ง “ng new lab4-angular-app”
2. ติดตั้ง Bootstrap CSS Framework ด้วยการพิมพ์คำสั่ง “npm install bootstrap jquery popper.js”
3. สร้าง Component ใหม่ชื่อ “signup” สำหรับสร้าง Form ในการกรอกข้อมูลจากผู้ใช้งาน
4. เปิดไฟล์ app.module เพื่อนำเข้า ReactiveFormsModule มาใช้งานในแอปพลิเคชันที่สร้างขึ้น
5. แก้ไขไฟล์ signup.component.ts โดยนำเข้าโมดูล FormControl และสร้าง Form Control Instance ชื่อ “firstName” และ “lastName” เพื่อใช้เชื่อมโยงข้อมูลจาก Form ที่อยู่ใน Template ดังต่อไปนี้

```

TS signup.component.ts ×
src > app > components > signup > TS signup.component.ts > ...
1  import { Component, OnInit } from '@angular/core';
2  import { FormControl } from '@angular/forms';
3
4  @Component({
5    selector: 'app-signup',
6    templateUrl: './signup.component.html',
7    styleUrls: ['./signup.component.css']
8  })
9  export class SignupComponent implements OnInit {
10
11    firstName = new FormControl('');
12    lastName = new FormControl('');
13
14    constructor() { }
15
16    ngOnInit(): void {
17    }
18
19  }

```

6. แก้ไขไฟล์ signup.component.html โดยทำการเชื่อมโยง Form เข้ากับ Form Control Instance ที่สร้างเตรียมไว้จากขั้นตอนที่ 5 พร้อมกับแสดงผลลัพธ์ออกที่หน้า Template ดังต่อไปนี้

```

<> signup.component.html X
src > app > components > signup > <> signup.component.html > div.container > div.card > div.card-body > div.row > div
1 <div class="container" style="margin-top:60px">
2   <div class="card">
3     <div class="card-header">
4       Signup
5     </div>
6     <div class="card-body" style="margin-left:10px">
7       <div class="row">
8         <div class="col-md-7">
9
10        <form>
11          <div class="form-group">
12            <label>FirstName</label>
13            <input type="text" class="form-control" [formControl]='firstName'>
14          </div>
15          <div class="form-group">
16            <label>LastName</label>
17            <input type="text" class="form-control" [formControl]='lastName'>
18          </div>
19          <br/><br/>
20          <button type="submit" class="btn btn-primary">Submit</button>&nbsp;
21          <button type="reset" class="btn btn-danger">Reset</button>
22        </form>
23
24      </div>
25
26      <div class="col-md-5">
27        <div class="card">
28          <div class="card-body">
29            <h4 class="card-title">Display :</h4>
30            <p class="card-text">
31              FirstName: {{firstName.value}} <br/>
32              LastName: {{lastName.value}}
33            </p>
34          </div>
35        </div>
36      </div>
37    </div>
38  </div>
39 </div>
40 </div>
41

```

7. หน้าต่าง Command Prompt ทดสอบการทำงานของแอปพลิเคชันด้วยการพิมพ์คำสั่ง “ng serve --open”
8. เปิดเบราว์เซอร์ จากนั้นกรอก url คือ “http://localhost:4200” จะปรากฏที่หน้าเว็บเพจ โดยกรอกข้อมูลลงไป จะแสดงผลดังรูปต่อไปนี้

The screenshot shows a web browser window with the address bar displaying 'localhost:4200'. The page content includes a 'Signup' form. The form consists of two text input fields. The first field, labeled 'FirstName', contains the text 'Nuntawut'. The second field, labeled 'LastName', contains the text 'Kaoungku'. Below these fields are two buttons: a blue 'Submit' button and a red 'Reset' button. To the right of the input fields, there is a section titled 'Display :'. This section shows the data entered into the form: 'FirstName: Nuntawut' and 'LastName: Kaoungku'.

แบบฝึกปฏิบัติการที่ 4.2 Reactive Form: Form Group

1. แก้ไขไฟล์ `signup.component.ts` โดยนำเข้าโมดูล `FormGroup` และสร้าง `Form Control Instance` เพื่อใช้เชื่อมโยงข้อมูลจาก Form ที่อยู่ใน Template ดังต่อไปนี้

```
TS signup.component.ts ×
src > app > components > signup > TS signup.component.ts > ...
1  import { Component, OnInit } from '@angular/core';
2  import { FormControl, FormGroup } from '@angular/forms';
3
4  @Component({
5    selector: 'app-signup',
6    templateUrl: './signup.component.html',
7    styleUrls: ['./signup.component.css']
8  })
9
10 export class SignupComponent implements OnInit {
11
12   profileForm = new FormGroup({
13     title: new FormControl(''),
14     firstName: new FormControl(''),
15     lastName: new FormControl(''),
16     sex: new FormControl(''),
17     email: new FormControl(''),
18     address: new FormGroup({
19       street: new FormControl(''),
20       city: new FormControl(''),
21       state: new FormControl(''),
22       zip: new FormControl(''),
23     })
24   });
25
26   constructor() { }
27
28   ngOnInit(): void {
29
30   }
31
32 }
```

2. แก้ไขไฟล์ `signup.component.html` โดยทำการเชื่อมโยง Form เข้ากับ `Form Control Instance` ที่สร้างเตรียมไว้จากขั้นตอนที่ 1 พร้อมกับแสดงผลป้อนออกที่หน้า Template ดังต่อไปนี้

```

<form [formGroup]="profileForm">

  <div class="form-group">
    <label for="">Title</label>
    <select class="form-control" formControlName="title">
      <option value=""></option>
      <option value="Mr">Mr</option>
      <option value="Mrs">Mrs</option>
      <option value="Miss">Miss</option>
      <option value="Ms">Ms</option>
    </select>
  </div>

  <div class="form-group">
    <label>FirstName</label>
    <input type="text" class="form-control" formControlName='firstName'>
  </div>

  <div class="form-group">
    <label>LastName</label>
    <input type="text" class="form-control" formControlName='lastName'>
  </div>

  <div class="form-check form-check-inline">
    <label class="form-check-label">
      <input class="form-check-input" type="radio" name="sex" value="Male" formControlName='sex'> Male
    </label>
  </div>

  <div class="form-check form-check-inline">
    <label class="form-check-label">
      <input class="form-check-input" type="radio" name="sex" value="Female" formControlName='sex'> Female
    </label>
  </div>
  <br/><br/>

  <div class="form-group">
    <label>Email</label>
    <input type="email" class="form-control" formControlName='email'>
  </div>

  <div formGroupName="address">
    <h4>Address</h4>
    <div class="form-group">
      <label>Street</label>
      <input type="text" class="form-control" formControlName='street'>
    </div>
    <div class="form-group">
      <label>City</label>
      <input type="text" class="form-control" formControlName='city'>
    </div>
    <div class="form-group">
      <label>State</label>
      <input type="text" class="form-control" formControlName='state'>
    </div>
    <div class="form-group">
      <label>Zip</label>
      <input type="text" class="form-control" formControlName='zip'>
    </div>
  </div>

  <br/><br/>
  <button type="submit" class="btn btn-primary">Submit</button>&nbsp;
  <button type="reset" class="btn btn-danger">Reset</button>
</form>

</div>

<div class="col-md-5">
  <div class="card">
    <div class="card-body">
      <h4 class="card-title">Display :</h4>
      <p class="card-text">
        {{profileForm.value|json}}
      </p>
    </div>
  </div>
</div>
</div>

```

- เปิดเบราว์เซอร์ จากนั้นกรอก url คือ “http://localhost:4200” จะปรากฏที่หน้าเว็บเพจ โดยกรอกข้อมูลลงไป จะแสดงผลดังรูปต่อไปนี้

Signup

Title

Mr

FirstName

Nuntawut

LastName

Kaoungku

☒ Male ☐ Female

Email

nuntawut@sut.ac.th

Address

Street

University

City

Muang

State

Nakhon Ratchasima

Zip

30000

Display :

{ "title": "Mr", "firstName": "Nuntawut", "lastName": "Kaoungku", "sex": "Male", "email": "nuntawut@sut.ac.th", "address": { "street": "University", "city": "Muang", "state": "Nakhon Ratchasima", "zip": "30000" } }

Submit

Reset

แบบฝึกปฏิบัติการที่ 4.3 Reactive Form: Form Array

- แก้ไขไฟล์ signup.component.ts โดยนำเข้าโมดูล FormArray และสร้าง Form Control Instance เพื่อใช้เชื่อมโยงข้อมูลจาก Form ที่อยู่ใน Template ดังต่อไปนี้

```

TS signup.component.ts ×
src > app > components > signup > TS signup.component.ts > SignupComponent
1  import { Component, OnInit } from '@angular/core';
2  import { FormControl, FormGroup, FormArray } from '@angular/forms';
3
4  @Component({
5    selector: 'app-signup',
6    templateUrl: './signup.component.html',
7    styleUrls: ['./signup.component.css']
8  })
9
10 export class SignupComponent implements OnInit {
11
12   profileForm = new FormGroup({
13     title: new FormControl(''),
14     firstName: new FormControl(''),
15     lastName: new FormControl(''),
16     sex: new FormControl(''),
17     email: new FormControl(''),
18     address: new FormGroup({
19       street: new FormControl(''),
20       city: new FormControl(''),
21       state: new FormControl(''),
22       zip: new FormControl(''),
23     }),
24     aliases: new FormArray([
25       new FormControl('')
26     ])
27   });
28
29   get aliases() {
30     return this.profileForm.get('aliases') as FormArray;
31   }
32
33   constructor() { }
34
35   ngOnInit(): void {
36
37   }
38
39   addAlias() {
40     this.aliases.push(new FormControl(''));
41   }
42
43 }

```

- แก้ไขไฟล์ signup.component.html โดยทำการเชื่อมโยง Form เข้ากับ Form Control Instance ที่สร้างเตรียมไว้จากขั้นตอนที่ 5 พร้อมกับแสดงผลป้อนออกที่หน้า Template ดังต่อไปนี้

```

<div formArrayName="aliases">
  <h4>Aliases</h4>
  <button class="btn btn-success" (click)="addAlias()">Add Alias</button>

  <div class="form-group" *ngFor="let alias of aliases.controls; let i=index">
    <label>Alias: {{i}}</label>
    <input type="text" class="form-control" [formControlName]="i">
  </div>
</div>

```

- เปิดเบราว์เซอร์ จากนั้นกรอก url คือ “http://localhost:4200” จะปรากฏที่หน้าเว็บเพจ โดยกรอกข้อมูลลงไป จะแสดงผลดังรูปต่อไปนี้

Aliases

Add Alias

Alias: 0

B5111299

Alias: 1

Nuntawut

Alias: 2

Tum

SubmitReset

Display :
{ "title": "", "firstName": "", "lastName": "", "sex": "", "email": "", "address": { "street": "", "city": "", "state": "", "zip": "" }, "aliases": ["B5111299", "Nuntawut", "Tum"] }

แบบฝึกปฏิบัติการที่ 4.4 Reactive Form: Form validation

- แก้ไขไฟล์ signup.component.ts โดยนำเข้าโมดูล Validator และกำหนดเงื่อนไขเพื่อตรวจสอบความถูกต้องของข้อมูลที่กรอกเข้ามาจากผู้ใช้งานในส่วนของ email ดังต่อไปนี้

```
export class SignupComponent implements OnInit {  
  
  profileForm = new FormGroup({  
    title: new FormControl(''),  
    firstName: new FormControl(''),  
    lastName: new FormControl(''),  
    sex: new FormControl(''),  
    email: new FormControl('', [Validators.required, Validators.email]),  
    address: new FormGroup({  
      street: new FormControl(''),  
      city: new FormControl(''),  
      state: new FormControl(''),  
      zip: new FormControl(''),  
    }),  
    aliases: new FormArray([  
      new FormControl('')  
    ])  
  });  
  
  get email() { return this.profileForm.get('email'); }  
}
```

- แก้ไขไฟล์ signup.component.html ในส่วนของการกรอกข้อมูล email เพื่อดักจับ error ที่เกิดขึ้นจาก Validator ที่กำหนดในขั้นตอนที่ 1 และแสดงแจ้งเตือนออกที่ template ผ่าน directive ของ angular ดังต่อไปนี้

```

<div class="form-group">
  <label>Email</label>
  <input type="email" class="form-control" formControlName='email'>
  <div *ngIf="email.invalid && (email.dirty || email.touched)"
    class="alert alert-danger">
    <div *ngIf="email.errors.required">
      Email is required.
    </div>
    <div *ngIf="email.errors.email">
      Your email is incorrect!!
    </div>
  </div>
</div>

```

3. เปิดเบราว์เซอร์ จากนั้นกรอก url คือ “http://localhost:4200” จะปรากฏที่หน้าเว็บเพจ โดยกรอกข้อมูลลงไป จะแสดงผลดังรูปต่อไปนี้

<p><input type="radio"/> Male <input type="radio"/> Female</p> <p>Email</p> <input type="text"/> <div style="background-color: #f8d7da; padding: 2px; margin-top: 5px;">Email is required.</div> <p>Address</p> <p>Street</p> <input type="text"/>	<p><input type="radio"/> Male <input type="radio"/> Female</p> <p>Email</p> <input type="text" value="nun"/> <div style="background-color: #f8d7da; padding: 2px; margin-top: 5px;">Your email is incorrect!!</div> <p>Address</p> <p>Street</p> <input type="text"/>
---	--

4. ให้ นศ. เพิ่มในส่วนของ form สำหรับกรอกรหัส นศ. (student id.) หลังจากทำแบบฝึกปฏิบัติการที่ 4.4 เสร็จโดยจะต้องตรวจสอบข้อมูล (Validation) ดังนี้

```

stdid: new FormControl('', [Validators.required, Validators.pattern('B[0-9]{7}')]),

```