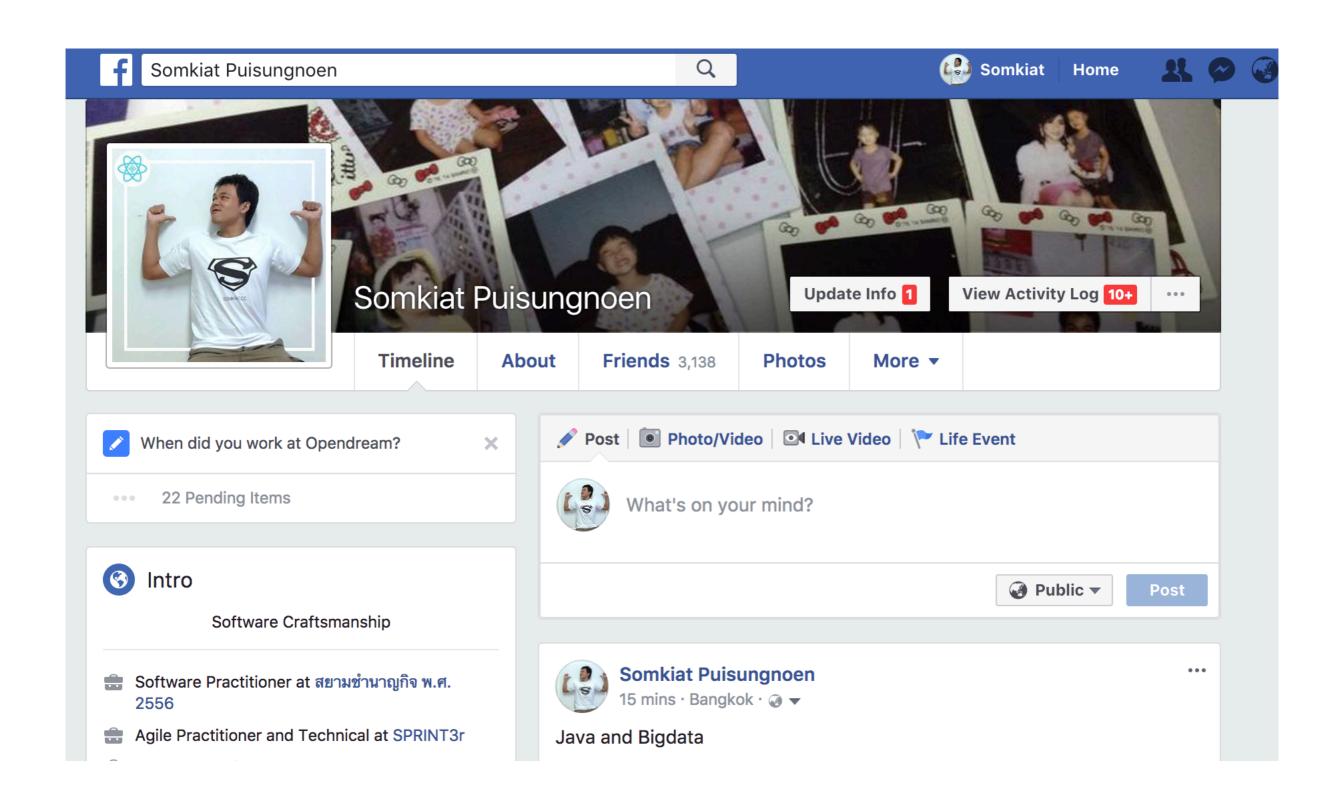
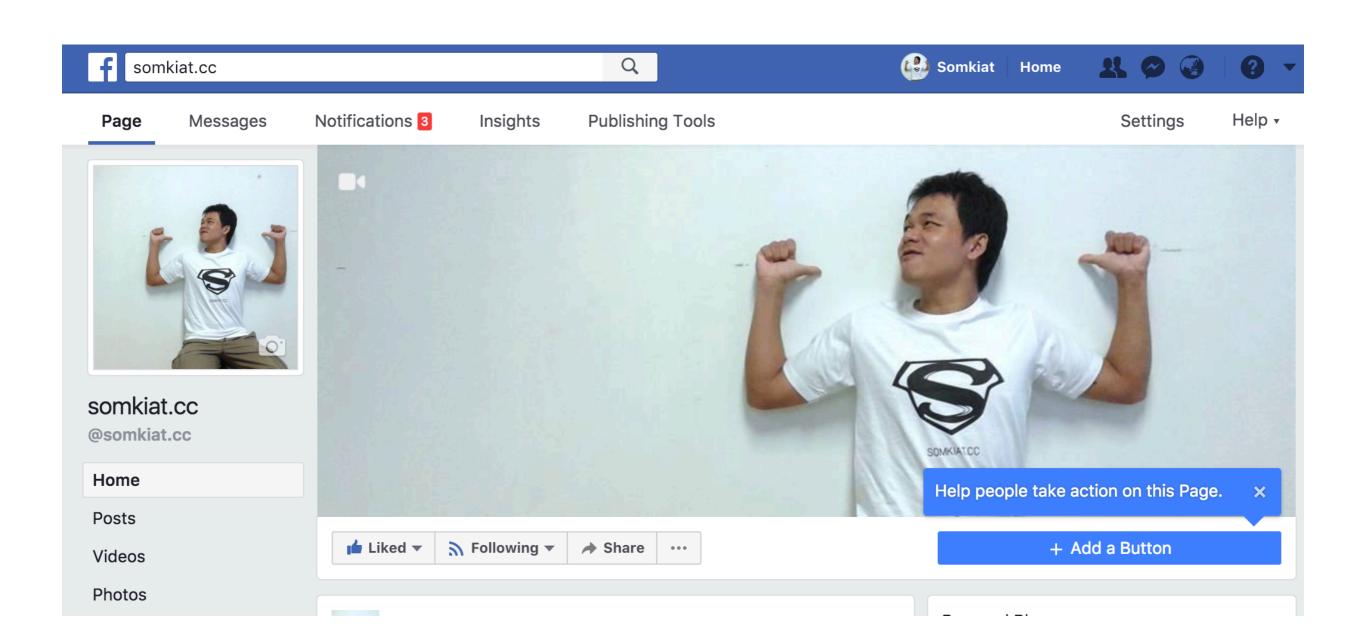


Workshop











Workshop



Feature 1

Add data to array

First name: firstname

Last name: lastname

Save

Count = 1/5

- firstname lastname



User.component.ts

```
users = [];
show = true;
selected = 0;
saveData(): void {
  const fullName = `${this.firstName} ${this.lastName}`;
  this.users.push({ name: fullName, status: false });
onSelect(index: number): void {
  this.users[index].status = !this.users[index].status;
  this.selected = this.users.filter((u) \Rightarrow u.status).length;
```



User.component.html



User.component.css

```
li:hover {
   cursor: pointer;
}
li.selected {
   text-decoration: line-through;
}
```



Reactive Forms Module

https://angular.io/guide/reactive-forms



app.module.ts

```
import { ReactiveFormsModule } from '@angular/forms';

...

@NgModule({
   declarations: [AppComponent, UsersComponent],
   imports: [BrowserModule, AppRoutingModule, ReactiveFormsModule],
   providers: [],
   bootstrap: [AppComponent],
})
export class AppModule {}
```



User.component.html

```
<form [formGroup]="loginForm" (ngSubmit)="saveData()">
 First name:
  <input formControlName="firstName" />
  <br />
 Last name:
  <input formControlName="lastName" />
  <br />
  <button>Save</button>
  <br />
</form>
```



User.component.ts (1)

```
import { FormGroup, FormBuilder } from '@angular/forms';
export class UsersComponent implements OnInit {
 loginForm: FormGroup;
 constructor(private formBuilder: FormBuilder) {}
 ngOnInit(): void {
    this.loginForm = this.formBuilder.group({
     firstName: [''],
      lastName: [''],
    });
```



User.component.ts (2)

```
get f() {
    return this.loginForm.controls;
}

saveData(): void {
    const fullName = `${this.f.firstName.value} $
{this.f.lastName.value}`;

    this.users.push({ name: fullName, status: false });
}
```



Validation with form

https://angular.io/guide/reactive-forms

