

MŲC TIÊU:

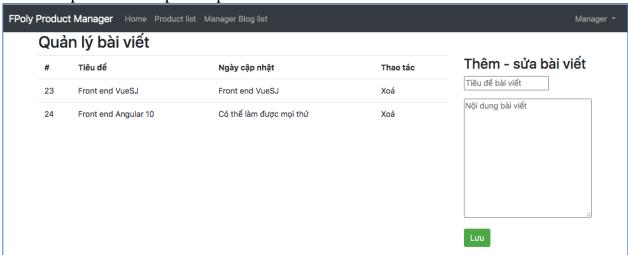
Kết thúc bài thực hành này bạn có khả năng

- ✓ Angular giao tiếp với Backend
- ✓ Sử dụng module http service để làm việc với Backend
- ✓ Tạo và sử dụng service cho yêu cầu với http

PHẦNI

Bài 1 (2 điểm)

Sử dụng Api của học phần NodeJs và Http request Angular thực hiện hiển thị danh sách bài viết – post trên component post



Hướng dẫn

Tạo component post và thiết kế giao diện như mẫu Cài đặt Http module tại app.module.ts:

import {HttpClientModule} from '@angular/common/http';

imports: [

BrowserModule,

AppRoutingModule,

FormsModule.

ReactiveFormsModule.

HttpClientModule



1

Đọc danh sách bài viết với phương thức GET của đối tượng HttpClient

```
ngOnInit(): void {
    this.getAllPost();
}
getAllPost(){
    this.httpService.get(this.url).subscribe(data=>{
        this.posts=data.posts;
    })
}
```

Hiển thị bài viết ra ngoài giao diện

```
#

Tiêu dề

Ngày cập nhật

Thao tác

*/tr>

{{post.postld}}

{{post.content}}

*{td>{{post.content}}

**td>**To**Pichen***

**To**Pichen**

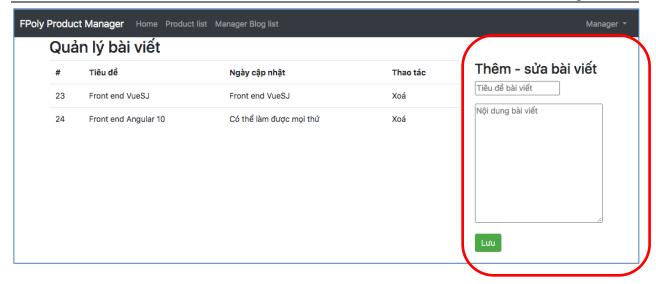
**To**Pich
```

Xem kết quả:localhost:4200/posts

Bài 2 (2 điểm)

Thực hiện thêm 1 bài viết mới





Hướng dẫn

Thực hiện binding dữ liệu 2 chiều trên form

Tại post.component.ts thực hiện code

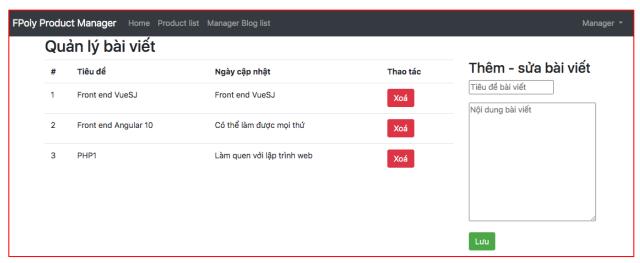
```
post:IPost;
onCreatePost(dataPost:IPost){
    this.httpService.post(this.url,dataPost).subscribe(p=>{
        console.log(p);
    })
}
addPost(){
    this.onCreatePost(this.post);
    this.getAllPost();
}
```



Kết quả: như hình trên

Bài 3 (2 điểm)

Thực hiện thao tác thiết kế nút xoá và code cho nút này



Hướng dẫn

Bổ sung phần Html như sau

```
=1">

{{i+1}}
{{i+1}}
{{ctd>{{ipost.title}}
{{td>}
{{post.content}}
{{td>}
{{button class="btn btn-danger" (click)="onDelete(post.postId)"> Xoá dbutton class="btn btn-danger" (click)="onDelete(post.postId)" (click)="onDelete(post.postId)"</br>
```

Hàm onDelete viết tại Component.ts:

```
onDelete(id:number){
  this.httpService.delete(this.url+id).subscribe(p=>{
    console.log(p);
    this.getAllPost();
  })
}
```

Chạy và kiểm tra kết quả



PHẦN II

Bài 4 (2 điểm)

Thực hiện các chức năng thêm, xoá, sửa post với service cho http request **Hướng dẫn:**

- Tao service

ng generate service post

- Code cho service post

```
import { Injectable } from '@angular/core';
 import { HttpClient } from '@angular/common/http';
 import { IPost } from './entities/post';
@Injectable({
 providedIn: 'root'
})
export class PostService {
url='http://127.0.0.1:3000/blog/posts/';
  constructor(private httpService:HttpClient) { }
  storePost(postData:IPost){
  this.httpService.post(this.url,postData).subscribe(p=>{
   console.log(p);
  })
  getAllPost(){
  return this.httpService.get(this.url);
  deletePost(id:number){
  return this.httpService.delete(this.url+id);
```

- Sử dụng service tại post.component.ts

```
ngOnInit(): void {
    this.getAllPost();

}
getAllPost(){
    this.postService.getAllPost().subscribe(data=>{
        this.posts=data.posts;
        console.log(this.posts);
    })
}
addPost(){
    this.postService.storePost(this.post);
}
```



```
onDelete(id:number){
  this.postService.deletePost(id).subscribe(p=>{
    console.log(p);
    this.getAllPost();
})
}
```

- Chạy và kiểm tra kết quả xử lý khi nhấn submit với console của trình duyệt

Bài 5 (2 điểm)

Thực hiện Quản lý lỗi xảy ra khi thực hiện các thao tác thêm, xoá, sửa

Xây dựng trạng thái loading... khi ứng dụng đang tải dữ liệu

```
=1 && !isFetching" <br/>
<br/>
{{i+1}}{{i+1}}<{td>{{post.title}}{{post.content}}<button class="btn btn-danger" (click)="onDelete('1234')">Xoá</button><br/>
p *ngl/='isFetching'- Loading....
```

```
getAllPost(){
    this.isFetching=true;
    this.postService.getAllPost().subscribe(data=>{
        this.isFetching=false;
        this.posts=data.posts;
        console.log(this.posts);
    },error=>{
        this.error=error.message;
    })
}
```

*** Yêu cầu nộp bài:

SV nén file (hoặc share thư mục google drive) bao gồm các yêu cầu đã thực hiện trên, nộp LMS đúng thời gian quy định của giảng viên. KHÔNG NỘP BÀI COI NHƯ KHÔNG CÓ ĐIỂM.

--- Hết ---