TRƯỜNG ĐẠI HỌC KINH TẾ

**KHOA THỐNG KÊ – TIN HỌC**

–––––––––––––––––––––––––––––––

**BÁO CÁO THỰC TẬP NGHỀ NGHIỆP**

**TÊN ĐỀ TÀI**

**Xây dựng website quản lý sinh viên**

**Sinh viên thực hiện:**

Trương Công Minh Lớp : 44K14

**Giảng viên hướng dẫn:** Cao Thị Nhâm

**Đà Nẵng, 7/2021**

**LỜI CẢM ƠN**

**LỜI CAM ĐOAN**

Quy định:

Soạn thảo trên trang A4 (trang dọc, lề trái: 3.5cm; trên, phải, dưới: 2.5 cm), Font Times New Roman, canh đều 2 bên, size 13, cách dòng 1.5, cách đoạn trên 6pt, cách đoạn dưới 3pt; hình và bảng soạn theo caption, chèn trích dẫn chéo (Cross-reference) cho bảng và hình; các danh mục hình, bảng, mục lục làm tự động; đánh số trang như file mẫu (bìa không có số trang, danh mục + mục lục số trang theo i, ii, iii…, nội dung chính theo 1,2,3…)

# MỤC LỤC

[MỤC LỤC ii](#_Toc74643748)

[LỜI MỞ ĐẦU 1](#_Toc74643749)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 2](#_Toc74643750)

[1.1. Tổng quan về .NET framework 2](#_Toc74643751)

[1.1.1. Mục 1.1.1 2](#_Toc74643752)

[1.1.2. Mục 1.1.2 2](#_Toc74643753)

[1.2. Tổng quan về Windows Forms 2](#_Toc74643754)

[1.3. Tổng quan về hệ quản trị cơ sở dữ liệu Microsoft SQL Server 2](#_Toc74643755)

[CHƯƠNG 2. phân tích và thiết kế ứng dụng 3](#_Toc74643756)

[2.1. Mô tả bài toán 3](#_Toc74643757)

[2.2. Thiết kế các chức năng 3](#_Toc74643758)

[2.3. Thiết kế cơ sở dữ liệu 3](#_Toc74643759)

[CHƯƠNG 3. PHÁT TRIỂN ỨNG DỤNG 4](#_Toc74643760)

[3.1. Sơ đồ chức năng 4](#_Toc74643761)

[3.2. Thông tin chi tiết các form làm việc 4](#_Toc74643762)

[3.2.1. Form X 4](#_Toc74643763)

[3.2.2. Form Y 4](#_Toc74643764)

[CHƯƠNG 4. KẾT QUẢ 5](#_Toc74643765)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 6](#_Toc74643766)

[TÀI LIỆU THAM KHẢO 7](#_Toc74643767)

[PHỤ LỤC (nếu có) 8](#_Toc74643768)

DANH MỤC HÌNH ẢNH

[Hình 1.1 Ngôn ngữ lập trình Python 2](#_Toc74235469)

[Hình 4.1 Kiến trúc của mô hình RNN 5](#_Toc74235470)

DANH MỤC BẢNG BIỂU

[Bảng 4.1 Kiến trúc 5](#_Toc74235471)

DANH MỤC CÁC TỪ VIẾT TẮT

**AI** : Artificial Intelligence

**…**

# LỜI MỞ ĐẦU

1. **Mục tiêu của đề tài**

* Nghiên cứu ngôn ngữ lập trình TypeScript
* Nghiên cứu framework xây dựng giao diện website Angular 12
* Nghiên cứu REST API và Json Server
* Xây dựng ứng dụng web Quản lý sinh viên

1. **Nhiệm vụ của đề tài**

* Nghiên cứu ngôn ngữ lập trình TypeScript
* Nghiên cứu framework xây dựng giao diện website Angular 12
* Nghiên cứu REST API và Json Server
* Phân tích bài toán quản lý sinh viên
* Phát triển ứng dụng để quản lý sinh viên

1. **Kết cấu của đề tài**

Đề tài được tổ chức gồm 4 chương nội dung và phần kết luận.

* **Chương 1**: Cơ sở lý thuyết
* **Chương 2**: Phân tích và thiết kế ứng dụng
* **Chương 3**: Phát triển ứng dụng
* **Chương 4**: Kết quả
* Kết luận và hướng phát triển

# CƠ SỞ LÝ THUYẾT

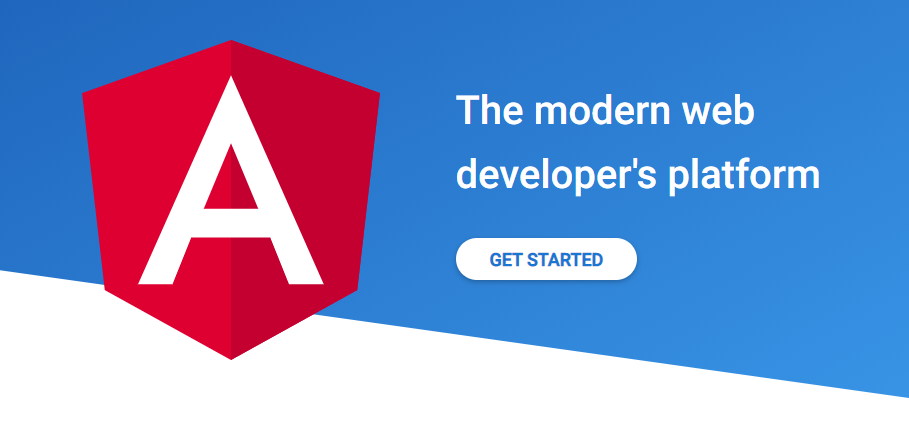
## Tổng quan về Angular framework

### Giới thiệu

Angular là một JavaScript framework dùng để viết giao diện web (Front-end), được tạo ra để xây dựng các ứng dụng web động (dynamic web app), nó thường được sử dụng để tạo ra các ứng dụng một trang (Single Page Application - SPA). Và hoạt động dựa trên việc mở rộng thêm các thuộc tính (attribute) cho các thẻ của HTML (Các thuộc tính theo quy tắc của Angular).

Angular là một nền tảng miễn phí và được hàng ngàn lập trình viên trên thế giới ưa chuộng. Hiểu đơn giản, Angular là một khung làm việc của JavaScript MVC phía máy khách (client) nhằm phát triển ứng dụng web động.

AngularJS là từ dùng để nói về Angular 1 (ra đời năm 2009), được viết bằng JavaScript. Angular là từ gọi chung cho Angular 2 trở lên (ra đời năm 2016), được viết bằng TypeScript – phiên bản nâng cao của JavaScript. Angular được thay đổi rất nhiều từ AngularJS. Angular đã thiết kế lại từ đầu nên có nhiều khái niệm đã thay đổi từ AngularJS. Kiến trúc của Angular và AngularJS hoàn toàn khác nhau.



Hình 1.1 Ngôn ngữ lập trình Python

### Tại sao nên dùng Angular

Angular – ứng dụng là một tập hợp những component (thành phần) và một vài services (dịch vụ) cung cấp chức năng trên những component đó.

* Ứng dụng được giảm tối đa kích thước và tăng tối đa hiệu suất.
* HTML linh hoạt hơn.
* Code HTML mạnh mẽ hơn với những đặc trưng như IF , FOR, LOCAL VARIABLES, …
* Dễ dàng hiển thị các field từ data model của website và theo dõi những thay đổi, cập nhật lại từ người dùng nhờ binding data.
* Xây dựng và tái sử dụng nội dung nhờ vào những khối module độc lập.
* Nhanh chóng giải quyết các bài toán logic nhờ back- end service hỗ trợ giao tiếp

## Tổng quan về TypeScript

TypeScript là một phiên bản cao hơn của JavaScript, được thiết kế để xây dựng các ứng dụng lớn và phức tạp. Nó kế thừa nhiều khái niệm từ Java và C#, TypeScript là ngôn ngữ tĩnh (Static typed) có nghĩa là nó nghiêm ngặt và có trật tự trái ngược với free-type. Nó còn được bổ sung thêm lớp hướng đối tượng mà điều này không có ở Javascript.

*Ứu điểm của TypeScript:*

* Dễ dàng hơn trong phát triển các dự án lớn, được hỗ trợ bởi các Javascript Framework lớn.
* Hầu hết các cú pháp hướng đối tượng đều được hỗ trợ bởi Typescript như kế thừa, đóng gói, constructor, abstract, interface, implement, override…v.v
* Cách tổ chức code rõ ràng hơn, hỗ trợ cơ chế giúp kiến trúc hệ thống code hướng module, hỗ trợ namespace, giúp xây dựng các hệ thống lớn nơi mà nhiều lập trình viên có thể làm việc cùng nhau một cách dễ dàng hơn.
* Hỗ trợ các tính năng mới nhất của Javascript. TypeScript luôn đảm bảo việc sử dụng đầy đủ các kỹ thuật mới nhất của Javascript, ví dụ như version hiện tại là ECMAScript 2015 (ES6).
* Một lợi thế của Typescript nữa là mã nguồn mở vì vậy nó miễn phí và có cộng đồng hỗ trợ rất lớn.
* Với static typing, code viết bằng TypeScript dễ dự đoán hơn, và dễ debug hơn.

## Tổng quan về REST API và Json Server

1. ***REST API***

API (Application Programming Interface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như JSON hay XML.

REST (REpresentational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu

RESTful API là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile…) khác nhau giao tiếp với nhau. Chức năng quan trọng nhất của REST là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE…) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một RESTful API.

1. ***Json Server***

# phân tích và thiết kế ứng dụng

## Mô tả bài toán

Diễn đạt lại nội dung mô tả bài toán ở đây

## Thiết kế các chức năng

Trình bày các chức năng của ứng dụng ở đây

## Thiết kế cơ sở dữ liệu

Trình bày các bảng cơ sở dữ liệu ở đây

# PHÁT TRIỂN ỨNG DỤNG

## Sơ đồ chức năng

Trong này vẽ sơ đồ các form có trong ứng dụng

## Thông tin chi tiết các form làm việc

### Form X

Trong này giới thiệu chức năng, các mục có trong màn hình

### Form Y

Trong này giới thiệu chức năng, các mục có trong màn hình

# KẾT QUẢ

Phần này trình bày kết quả đạt được

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Đề tài đã thực hiện được …

# TÀI LIỆU THAM KHẢO

1. Phan, D.-V., et al., *Liver cancer prediction in a viral hepatitis cohort: A deep learning approach.* International Journal of Cancer, 2020. **147**(10): p. 2871-2878.

2. Phan, D.V., et al., *Deep learning approaches for sleep disorder prediction in an asthma cohort.* J Asthma, 2020: p. 1-9.

Tên tác giả, Tên tác giả… *Tên công trình*. Nơi xuất bản, năm, trang

# PHỤ LỤC (nếu có)