

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
KHOA CÔNG NGHỆ PHẦN MỀM



BÁO CÁO THỰC TẬP

LẬP TRÌNH BACKEND với ASP.NET API

Công ty thực tập : KMS Technology VN

Người phụ trách :

Thực tập sinh : Phùng Thanh Tú

TP. Hồ Chí Minh, tháng 12 năm 2022

## LỜI MỞ ĐẦU

Theo số liệu thống kê từ các trang tuyển dụng, đồng thời từ chính trải nghiệm cá nhân của bản thân em, nhu cầu về lập trình ứng dụng Web, đặc biệt là lập trình Backend hiện tại đang tăng cao hơn bao giờ hết. Trong thời kỳ mọi thứ đều có thể được đưa lên trình duyệt, ai cũng muốn có một trang Web cho mình, thì nguồn nhân lực về mảng Web, đặc biệt là nguồn nhân lực chất lượng cao hiện tại vẫn chưa đáp ứng đủ. Bằng chứng là thông qua các trang chuyên về tuyển dụng như Linkedin, Upwork, thì luôn có rất nhiều những tin tuyển dụng về web xuất hiện.

Đặc biệt là ASP.NET. Đây là công nghệ với nhu cầu tuyển dụng nhiều nhất tại Việt Nam. Bất chấp tình hình sa thải nhân viên ở các nước phát triển như Mỹ hay Singapore thì tại Việt Nam hiện nay nhu cầu lập trình viên C# - ASP.NET vẫn không hề có dấu hiệu đi xuống mà ngược lại còn đang tăng.

Năm bắt tinh hình trên, cộng với vốn kiến thức quý giá tích lũy được sau ba năm mài dũa dưới mái Trường Đại học Công nghệ Thông tin, em đã đặt sự nghiệp của mình là trở thành Web Developer. Và để tích lũy thêm kinh nghiệm thực tế từ doanh nghiệp. Em đã quyết định chọn KMS Technology VN - một môi trường năng động, hiện đại, chuyên nghiệp - làm điểm đến đầu tiên cho quá trình xây dựng sự nghiệp của em.

## LỜI CẢM ƠN

Trân trọng gửi lời cảm ơn Công ty KMS Technology VN đã tạo điều kiện cho em được góp công sức của mình để công hién cho Công ty và cũng là để học hỏi và phát triển bản thân thông qua môi trường chuyên nghiệp và cởi mở của Công ty.

Trong thời gian 2 tháng thử việc và 4 tháng làm dự án chính thức, em đã học hỏi được rất nhiều những kiến thức mà em chưa có cơ hội vận dụng khi còn đi học - những kiến thức nền tảng để xây dựng một ứng dụng Web hoàn chỉnh.

Đặc biệt cảm ơn anh *Truong Ngoc Duy*, đã training C#, .NET, và ASP.NET Core API, hướng dẫn, giúp đỡ cho chúng em tận tình cả những khó khăn trong công việc, đến những khó khăn việc làm quen với môi trường mới.

Cũng xin cảm ơn thầy cô trong khoa Công nghệ phần mềm đã nhiệt tình hỗ trợ, tạo điều kiện em làm bài báo cáo này.

Phùng Thanh Tú

TpHCM, ngày 26 tháng 12 năm 2022

---

Phùng Thanh Tú

## NHẬN XÉT CỦA KHOA

## MỤC LỤC

<b>Chương 1: Giới thiệu công ty thực tập</b>	<b>6</b>
1. Giới thiệu công ty KMS Technology VN	6
2. Sản phẩm của công ty	6
<b>Chương 2: Nội dung thực tập.</b>	<b>7</b>
1. Tìm hiểu công ty và các kỹ năng cơ bản trong công ty	7
2. Nghiên cứu kỹ thuật	7
a. Các công cụ làm việc	7
i. Git và Git Flow	7
ii. Docker	13
iii. Google Calendar	13
iv. Google Chat	13
v. Azure DevOps	14
b. C# và .NET	14
i. C#	14
ii. .NET	15
c. SQL Server và Entity Framework	17
i. Setup SQL Server bằng Docker	17
ii. Entity Framework	18
<b>Chương 3: Chi tiết về project</b>	<b>22</b>
1. Giới thiệu về dự án	22

## Chương 1: Giới thiệu công ty thực tập



### 1. Giới thiệu công ty KMS Technology VN

KMS Technology VN được thành lập vào tháng 1 năm 2009 tại US. Với 13 năm kinh nghiệm, hơn 1500 nhân viên, KMS đã khẳng định vị thế của mình bằng những sản phẩm được hàng triệu người sử dụng trên toàn cầu - và những hợp đồng hợp tác lâu dài với những đối tác lớn và tiềm năng..

### 2. Sản phẩm của công ty

- QA Symphony
- Katalon
- Kobiton
- Visily
- Groove

## Chương 2: Nội dung thực tập.

### 1. Tìm hiểu công ty và các kỹ năng cơ bản trong công ty

*Thời gian :* 1 ngày

*Nội dung :* Giới thiệu về công ty, cách tổ chức của công ty

Được nghe người phụ trách giới thiệu về công ty, quá trình thành lập và phát triển (như đã nhắc đến ở trên), quy trình làm việc từ cao xuống thấp, cách thức tổ chức của công ty .

Ngoài ra, thực tập sinh còn được giới thiệu về cách thức làm việc trong công ty như thời gian đi làm, các quy định cần phải tuân thủ, cách sử dụng email trong công việc...

*Kết quả :* Hiểu thêm về công ty KMS, quá trình thành lập và phát triển. Có thêm các kỹ năng về việc sử dụng email trong công việc, làm việc có kế hoạch, có kỷ luật, có trách nhiệm hơn.

### 2. Nghiên cứu kỹ thuật

#### a. Các công cụ làm việc

##### i. Git và Git Flow

Git là một hệ thống kiểm soát và quản lý mã nguồn. Nhờ hệ thống này mà một mã nguồn có thể được phát triển song song bởi nhiều lập trình viên khác nhau nhờ hệ thống quản lý phiên bản. Ngoài ra nhờ hệ thống này mà khi có lỗi hoặc thay đổi trong yêu cầu thì hoàn toàn có thể quay trở về phiên bản cũ

Git chắc chắn là một công cụ quá quen thuộc nên các câu lệnh cơ bản như pull, push, commit, checkout, ... em sẽ không đi sâu trong báo cáo này mà thay vào đó em sẽ chỉ tập trung vào những câu lệnh mà em cảm thấy tâm đắc trong quá trình thử việc này.

- **git status:** đây là câu lệnh mà trước khi vào doanh nghiệp em chưa biết đến, câu lệnh này cho phép hiển thị thông tin hiện tại của repository, về các file đã được staged, các file đã được track nhưng chưa stage, hoặc các file chưa được track

```

> ~/Documents/Git-Rep/pearn > feature/block± git status
On branch feature/block
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)

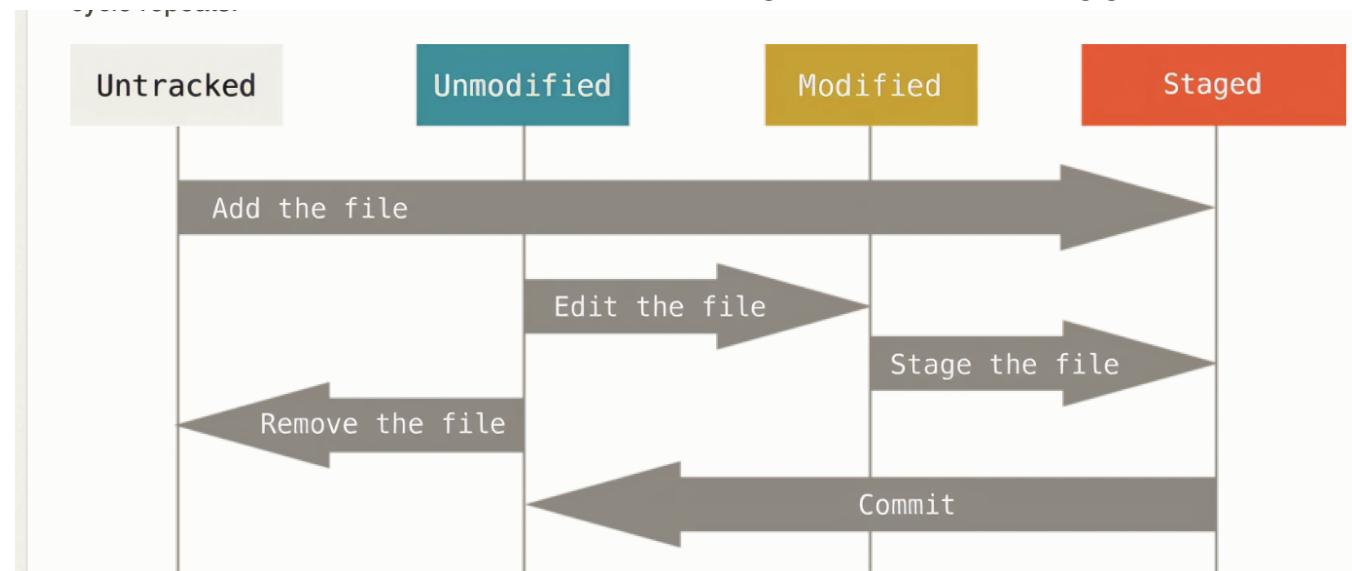
        modified:   backend/app.js
        modified:   backend/src/models/block.model.js
        modified:   backend/src/models/content.model.js
        modified:   backend/src/models/courses.model.js
        modified:   backend/src/services/block.service.js

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    backend/src/routes/block.js

no changes added to commit (use "git add" and/or "git commit -a")

```

Ở trong hình trên, khi ta dùng lệnh git status, ta có thể thấy được những thay đổi trong local đã được track. Sơ đồ sau khiến ta dễ hiểu hơn về các trạng thái của một file trong git



Khi ta dùng git status, ta có thể biết được các thái như thế nào, file nào đã được commit file nào chưa. **Đặc biệt là nếu ta vô tình commit những file thừa như node\_modules, bin, obj, ... do quên đưa chúng vào .gitignore thì khi check bằng git status thì ta sẽ biết và hạn chế việc commit nhầm file.**

- **git add**: đây là câu lệnh dùng để chuyển trạng thái của một file từ modified sang staged, tức là chuyển sang trạng thái sẵn sàng để commit. Câu lệnh này được xem là một câu lệnh cơ bản trong git, tuy nhiên nó hay bị dùng sai cách. Cụ thể, thường mọi người sẽ dùng git add . để staged toàn bộ các file điều này có thể dẫn đến việc commit nhầm file như đã nói ở trên. Nên kết hợp git status và git add để chỉ git add những file mà ta cần để commit.

```

no changes added to commit (use "git add" and/or "git commit -a")
❯ ~/Documents/Git-Rep/pearn ↵ feature/block± git add /src/block
fatal: Invalid path '/src': No such file or directory
✖ ❯ ~/Documents/Git-Rep/pearn ↵ feature/block± git add backend/src/models/block.model.js
❯ ~/Documents/Git-Rep/pearn ↵ feature/block± git add backend/src/services/block.service.js
❯ ~/Documents/Git-Rep/pearn ↵ feature/block± git add backend/src/routes/block.js
❯ ~/Documents/Git-Rep/pearn ↵ feature/block± git status
On branch feature/block
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   backend/src/models/block.model.js
    new file:   backend/src/routes/block.js
    modified:   backend/src/services/block.service.js

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   backend/app.js
    modified:   backend/src/models/content.model.js
    modified:   backend/src/models/courses.model.js

❯ ~/Documents/Git-Rep/pearn ↵ feature/block± -

```

Điều thú vị hơn ở đây là ta hoàn toàn có thể git add những file mà ta đang làm, tránh việc commit những file khác (của chức năng khác) vì chúng có thể gây conflict. Ví dụ như trên hình dựa vào tên branch ta có thể thấy chức năng ta đang làm liên quan đến block, do đó ta chỉ commit những file của chức năng block mà thôi

- **git config --global core.editor "code --wait"**: câu lệnh này cho phép ta chuyển editor commit message sang VS Code, thứ mà sẽ dễ nhìn và dễ xài hơn so với mặc định (vim). Còn editor commit message là gì thì câu lệnh **git commit -a** sẽ giải thích sau.
  - **git commit -a**: Cho phép edit commit message bằng editor mà ta đã chỉnh bằng lệnh git config

.git > COMMIT\_EDITMSG

```
1
2 # Please enter the commit message for your changes. Lines starting
3 # with '#' will be ignored, and an empty message aborts the commit.
4 #
5 # On branch feature/block
6 # Changes to be committed:
7 #   modified: backend/app.js
8 #   modified: backend/src/models/block.model.js
9 #   modified: backend/src/models/content.model.js
10 #  modified: backend/src/models/courses.model.js
11 # new file: backend/src/routes/block.js
12 # modified: backend/src/services/block.service.js
13 #
14
```

PROBLEMS DEBUG CONSOLE TERMINAL JUPYTER

git status

```
~/Documents/Git-Rep/pearn > feature/block > git status
On branch feature/block
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified: backend/src/models/block.model.js
    new file: backend/src/routes/block.js
    modified: backend/src/services/block.service.js

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified: backend/app.js
    modified: backend/src/models/content.model.js
    modified: backend/src/models/courses.model.js
```

git commit -a

```
hint: Waiting for your editor to close the file... [ ]
```

Ngoài ra thì trong commit message file còn hiện sẵn những file đã thay đổi giúp ta dễ dàng nắm bắt tình hình và những thứ ta đã làm trong commit đó

- ## - git log

```

commit 59c328c7ad263fc229bc86467a822fdbfd1ab2c1 (HEAD -> feature/block, origin/develop, develop)
Author: phungthanhtu <thanhtucqtuit1952@gmail.com>
Date:   Sun Dec 18 17:11:19 2022 +0700

    Hide confidential credentials in Mongoose
    connection string

commit 737d10209e2c60a956e11ae3353605d463f1a929
Author: phungthanhtu <thanhtucqtuit1952@gmail.com>
Date:   Sun Dec 18 16:16:53 2022 +0700

    Solve conflicts

commit 73925d309c84b1dc0053a956c064b5dc1a6d1eef
Author: phungthanhtu <thanhtucqtuit1952@gmail.com>
Date:   Sun Dec 18 16:08:02 2022 +0700

    Fix security vulnerabilities

commit b07314ccdcfc8ff82bbf673b2b587bb67998cf0
Author: phungthanhtu <thanhtucqtuit1952@gmail.com>
Date:   Sun Dec 18 15:59:17 2022 +0700

    Remove unnecessary imports

commit 780317821c7ef2003f70582cf502387e220c841b
Merge: 3728b51 cd88f81
Author: phungthanhtu <thanhtucqtuit1952@gmail.com>
Date:   Sun Dec 18 15:58:14 2022 +0700

    Merge branch 'feature/course-enhancement' into develop

commit cd88f815383520f1b4d26eb1b6255fbebe8c637
Author: phungthanhtu <thanhtucqtuit1952@gmail.com>
Date:   Sun Dec 18 15:57:11 2022 +0700

    New block model

commit 82e20c8e0d03fbaa669e449ec8468b4c5de1dd53
Author: phungthanhtu <thanhtucqtuit1952@gmail.com>
Date:   Sun Dec 18 15:56:20 2022 +0700

    Setup response Message

commit 45204632bcf25b063134e58271c6c43b09fac581
Author: phungthanhtu <thanhtucqtuit1952@gmail.com>
Date:   Fri Dec 16 17:32:45 2022 +0700

    Multiple students push and pull

commit 3728b5113dda844af85a1a7c7fd61c1c382b61b0
Author: phungthanhtu <thanhtucqtuit1952@gmail.com>
Date:   Thu Dec 15 14:54:26 2022 +0700

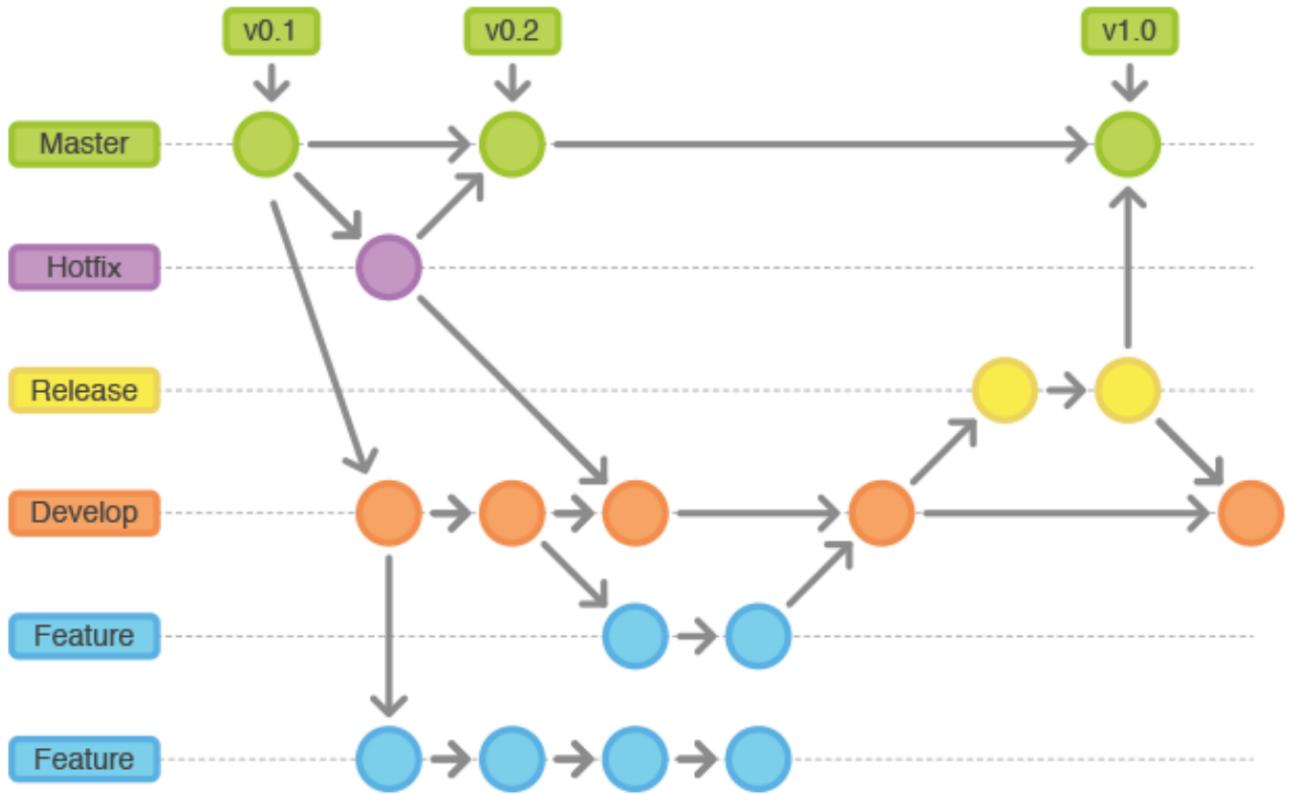
    fix delete endpoint

commit 4c71142797817edcc83583cb650754149daadb89

```

Câu lệnh này hiện lịch sử các commit, kết hợp với commit message, ta biết được chính xác ta đã commit những gì trên branch của mình

- **git flow:** đây là một git framework phát triển theo mô hình gitflow workflow. Mô hình này được tóm tắt trong sơ đồ sau.



Cụ thể, mô hình này chỉ ra cách thức setup các loại branch khác nhau và cách thức để merge chúng lại với nhau. Các loại branch này được chia ra thành các nhánh sau:

- master/main: đây là branch production, chứa mã nguồn của bản ứng dụng mà người dùng trực tiếp sử dụng. Commit của branch này được gắn các tag tương ứng với các phiên bản chính thức của ứng dụng.
- hotfix: đây là branch được base trên master, dùng để sửa lỗi nóng những lỗi khẩn cấp trên production
- release: đây là branch chứa các bản release của ứng dụng. Bản release này sẽ được team QA kiểm tra và kiểm thử trước khi được đưa lên production (merge vào main/master)
- develop: đây là branch mà dev sẽ làm việc nhiều nhất, đây là branch được base từ main/master, chứa toàn bộ lịch sử commit của ứng dụng. Đây là branch mà tất cả các branch feature sẽ merge vào
- feature: đây là một tập hợp các branch chức năng mà dev team làm việc. Các branch này được base ra từ develop và sau khi chức năng được hoàn thành, chúng

sẽ được merge vào lại develop. Cần phải tạo một pull request trước khi merge vào branch develop để team lead/ architect có thể review lại code trước khi merge.

#### *ii. Docker*

Docker là một công cụ cung cấp một môi trường ảo tách biệt với môi trường máy tính của chúng ta. Docker cho phép ta chạy bất kỳ ứng dụng nào miễn là có image của chúng trên docker hub.

Trong đợt thử việc này em sử dụng docker để setup một CSDL SQL Server mà không cần phải tải và cài đặt SQL Server vào máy. Lợi ích của việc này là một là cách thức mà ứng dụng kết nối với SQL Server sẽ y chang như kết nối với các Database trên Cloud (dùng SQL Authentication) chứ không phải Windows Authentication như so với kết nối SQL local trong máy. Mặt khác, CSDL SQL Server chạy trên Docker sẽ chạy được trên bất cứ hệ điều hành nào chạy được Docker (MacOS, Windows, Linux) chứ không giới hạn ở mỗi Windows như khi chạy SQL trên local.

#### *iii. Google Calendar*

Google Calendar là công cụ quản lý các sự kiện/ các cuộc họp dưới dạng lịch. Ngoài ra bên trong google calendar còn tích hợp sẵn họp bằng google meet nên mỗi khi có cuộc họp chỉ cần ấn vào link meet trên calendar là sẽ vào được phòng họp.

#### *iv. Google Chat*

Google Chat là một công cụ trò chuyện trên google - tuy nhiên có tích hợp thêm chức năng quản lý file thông qua Google Drive và quản lý Task dưới dạng to-do list.

The screenshot shows the Microsoft Teams application interface. On the left, there's a sidebar with sections for Chat, Spaces, and Meet. The main area is titled 'Team Đồ án' and shows a list of files under the 'Files' tab. The files listed are:

File	Posted by	Date posted
Members time management	Tú Phùng Thanh	Wednesday, Dec 7
Pearn Development Timeline	Tú Phùng Thanh	Tuesday, Dec 6
dotnet development timeline	Tú Phùng Thanh	Tuesday, Dec 6

### v. Azure DevOps

Azure DevOps là một bộ công cụ tổng hợp được Microsoft phát triển dành cho doanh nghiệp. Đây là một công cụ vô cùng mạnh, với chức năng bao phủ từ quản lý dự án (giống Jira), quản lý mã nguồn (của GitLab), cho đến quản lý CI/CD, Test Plan, thậm chí là quản lý các Tài liệu với Word, Excel, .... Điều đặc biệt là tất cả chúng đều nằm chung trong một Workspace duy nhất và toàn bộ các chức năng kết hợp với nhau một cách hoàn hảo hơn nhiều so với khi dùng kết hợp nhiều ứng dụng.

#### b. C# và .NET

##### i. C#

C# là một ngôn ngữ lập trình được các kỹ sư Microsoft phát triển vào những năm 2000. Trải qua quá trình phát triển và không ngừng cải tiến, đến nay C# được ứng dụng rộng rãi ở rất nhiều mảng, từ di động, nhúng đến web và game.

C# kế thừa những ưu điểm mà Java có - hướng đối tượng, kế thừa, interface, generic, ... Ngoài ra C# còn có những tính năng đặc trưng của JavaScript - await.

Đến phiên bản C# 10, C# đã hỗ trợ top-level statement, cho phép loại bỏ những cú pháp boilerplate như namespace, hàm Main, ... trong file Program.cs. Điều này được học hỏi từ ngôn ngữ python.

Có thể nói đội ngũ phát triển của C# rất hay cập nhật những tính năng thú vị mà các ngôn ngữ khác có. Điều này khiến cho trải nghiệm viết code trên C# trở nên tối ưu và dễ dàng hơn rất nhiều.

C# cũng chính là xương sống của .NET.

## *ii. .NET*

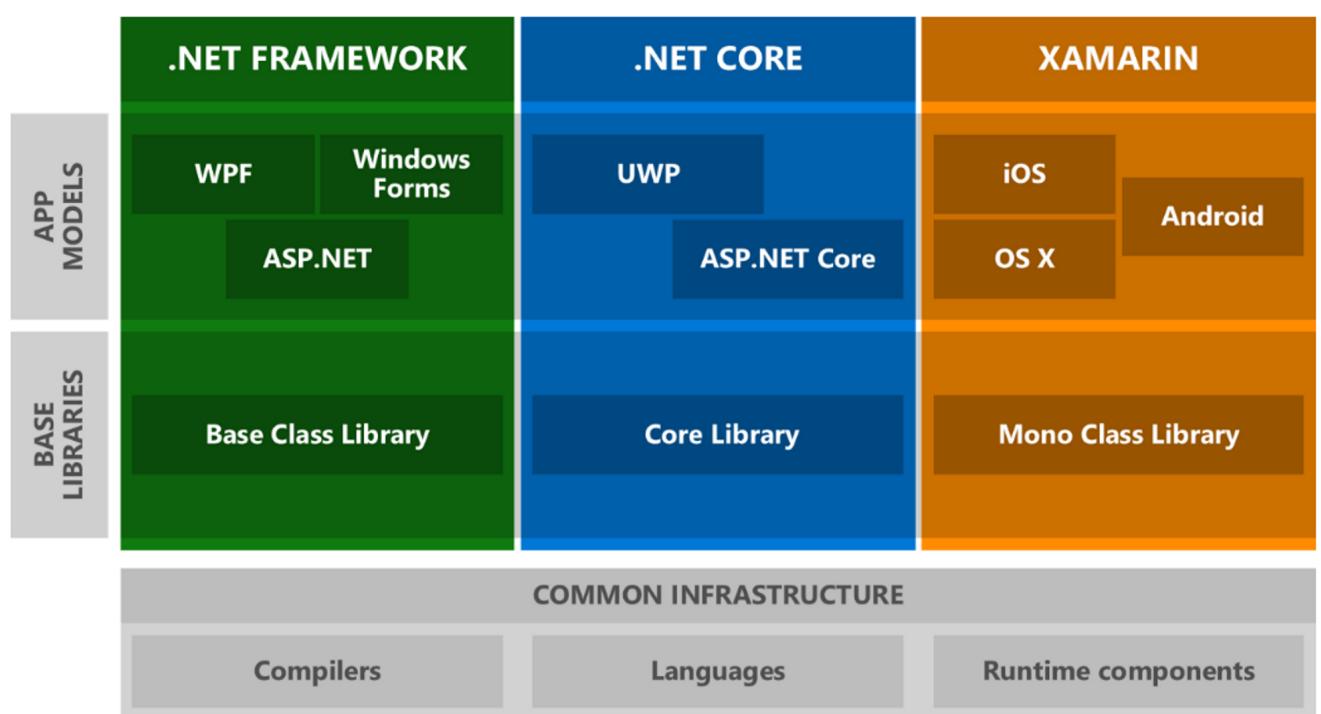
### *1. .NET là gì*

.NET là viết tắt của Network Enabled Technologies.

.NET đã từng có 2 phiên bản .NET Framework và .NET Core. Tuy nhiên sau phiên bản .NET Framework 4.8 và .NET Core 3.1 thì chúng đã “hợp nhất” lại thành một phiên bản duy nhất với tên gọi .NET (.NET 5). Hiện tại phiên bản LTS là .NET 6 còn phiên bản mới nhất đã ra mắt đến .NET 7.

.NET được sử dụng để xây dựng các ứng dụng chạy được trên nhiều nền tảng khác nhau.

Dưới đây là sơ đồ của .NET



### *2. Lịch sử .NET*

- 2002 -.NET Framework 1.0: giới thiệu CLR (Common Language Runtime).
- 2003 - .NET Framework 1.1: bổ sung thêm Mobile ASP.NET, hỗ trợ ODBC và Oracle.
- 2005 - .NET Framework 2.0: hỗ trợ ASP.NET, cập nhật thêm API, hỗ trợ generics
- 2006 - .NET Framework 3.0: bổ sung WPF, WF, WCS

- 2007 - .NET Framework 3.5: bổ sung LINQ, ASP.NET 3.5, REST, EntityFramework và hỗ trợ tính năng mới từ C# - lambda và expression tree
- 2009 - .NET Framework 4.0: Parallel LINQ, Task Parallel Library, hỗ trợ đa luồng và bắt đồng bộ (asynchronous)
- 2012 - .NET Framework 4.5: Bổ sung .NET API cho Store/UWP app, task-based async model
- 2016 - phiên bản .NET Core đầu tiên - .NET Core 1.0: đây là bản được xây dựng lại từ đầu với mã nguồn mở, đa nền tảng, có hỗ trợ CLI tools, ASP.NET Core, và Xamarin.Forms
- 2017 - .NET Core 1.1: Hỗ trợ Linux, cải thiện ASP.NET Core cho Kestrel server. Hỗ trợ Azure, EF Core, SQL 2016 và các API mới
- 2018 - tháng 5 - .NET Core 2.1: LTS, cải thiện đáng kể hiệu năng, hỗ trợ Ubuntu 18.04, Alpine, ARM32
- 2018 - tháng 12 - .NET Core 2.2: Cải thiện hiệu năng, hỗ trợ ARM32 Windows, và Azure Active Directory cho SQL Client
- 2019 là năm mà .NET Framework nhận được bản cập nhật cuối cùng - .NET Framework 4.8
- Ngoài ra vào năm 2019, .NET Core 3.0 ra đời với hiệu năng được cải thiện đáng kể, garbage collector tốn ít bộ nhớ hơn, hỗ trợ cả WPF và WinForms (nhưng chỉ trên Windows). Hỗ trợ Raspberry Pi và các chip ARM.
- 2020 - tháng 1 - .NET Core 3.1 là phiên bản .NET Core cuối cùng và là phiên bản LTS.
- 2020 - tháng 11 - .NET 5 ra đời, là một nền tảng thống nhất có thể chạy trên mọi thiết bị. Hỗ trợ Windows, Linux, MacOS, iOS, Android, tvOS, watchOS, WebAssembly, .... Ở thời điểm này không còn .NET Framework hay .NET Core nữa mà chỉ còn nền tảng duy nhất là .NET
- Tại thời điểm tìm hiểu báo cáo, .NET 6 LTS đã ra đời và đến hiện tại, .NET 7 đã ra mắt.

### **3. Ưu điểm và Nhược điểm .NET**

- **Ưu điểm:**
  - Đa chức năng, đa nền tảng
  - Mạnh mẽ, hỗ trợ tốt OOP, thích hợp để viết các ứng dụng lớn với khả năng maintain và mở rộng cao
  - Hỗ trợ đa luồng, generic, interface, attribute, linq, garbage collection, ....
- **Nhược điểm:**
  - Nặng, chiếm dung lượng rất lớn (> 20GB)
  - Vì là một công nghệ phổ biến và có tuổi đời cao nên số lượng lập trình viên nhiều dẫn đến tính cạnh tranh cao trong ngành

- Cộng đồng tương đối thiếu tương tác, trang community của Microsoft khá vắng vẻ, chủ yếu các cuộc thảo luận nằm trên StackOverflow
- Để tận dụng được với chất lượng tốt nhất thì phải tích hợp thêm các dịch vụ trả phí của Microsoft như Azure Cloud, ...

### c. *SQL Server và Entity Framework*

#### i. *Setup SQL Server bằng Docker*

- Cài đặt Docker Desktop theo hướng dẫn từ trang web chính thức của Docker <https://www.docker.com/products/docker-desktop/>
- Sau khi cài đặt, tạo một file docker-compose.yml và edit file đó như sau:

```

1  version: "3.8"
2
3  services:
4    mssql:
5      image: "mcr.microsoft.com/mssql/server:2019-latest"
6      container_name: mssql          # tên container
7      restart: always
8      user: root
9      hostname: mssql
10     environment:
11       SA_PASSWORD: Superadmin123456!1      # Thiết lập password
12       ACCEPT_EULA: Y
13     volumes:
14       - mssql:/var/opt/mssql
15     ports:
16       - "1433:1433"                      # cổng kết nối
17   volumes:
18     mssql:

```

- Thay đổi mật khẩu SA cho phù hợp
- cd vào folder chứa file docker-compose vừa tạo, sau đó chạy lệnh sau:

```
docker-compose up -f .\docker-compose.yml
```

Sau khi chạy xong, kết nối với CSDL dùng những thông tin sau:

The screenshot shows the 'Connection type' dropdown set to 'Microsoft SQL Server'. Below it, there are two radio buttons: 'Parameters' (selected) and 'Connection String'. The 'Server \*' field contains '127.0.0.1,1433'. The 'Authentication type' dropdown is set to 'SQL Login'. The 'User name \*' field contains 'SA' and the 'Password' field contains a masked password. A checked checkbox labeled 'Remember password' is present. The 'Database' dropdown is set to 'Master'. The 'Server group' dropdown is set to '<Default>'. An 'Advanced...' button is located below these fields. At the bottom, there are 'Connect' and 'Cancel' buttons.

Nếu kết nối thành công, tức là CSDL của chúng ta đã hoạt động, khi đó ta có thể sử dụng nó để gọi query như bình thường.

### *ii. Entity Framework*

Entity Framework là một ORM mã nguồn mở cho các ứng dụng .NET được Microsoft hỗ trợ. Entity Framework là một phần của .NET và là một open-source framework.

Entity Framework cho phép truy cập và tương tác với CSDL thông qua các object (còn gọi là Entity) nhằm chuyển đổi dữ liệu từ các object sang các câu truy vấn Query mà không cần phải gọi trực tiếp Query đến CSDL.

### *iii. ASP.NET Core*

ASP.NET Core là một phần của .NET. Đây là một nền tảng phát triển Web, dùng để tạo một trang web, hoặc một ứng dụng Web. Ngoài ra ASP.NET Core còn cho phép phát triển ứng dụng Backend thông qua ASP.NET Core API.

### *d. Unit Testing với xUnit và Moq*

#### *i. Unit Test*

Unit Test là một kỹ thuật white-box Testing khi ta kiểm thử một đơn vị riêng lẻ của hệ thống. Đơn vị riêng lẻ này có thể là một hàm, một phương thức, một lớp hoặc một Module riêng lẻ. Mục đích của việc kiểm thử này là để Debug và xác nhận rằng đơn vị đó hoạt động đúng như mong đợi.

Unit Test cô lập mã nguồn và xác minh tính đúng đắn của nó.

Mặc dù là một kỹ thuật testing, Unit Test được viết bởi các Dev chứ không phải Tester. Trong dự án thực tế, để vượt qua các đợt kiểm tra QA và đảm bảo mã nguồn đã được test (test coverage) thì Dev phải viết Unit Test cho mã nguồn của mình

### *ii. xUnit*

xUnit là một Unit Testing framework cung cấp công cụ để viết Unit Test, đồng thời chạy và debug dễ dàng cho các ứng dụng C# và .NET. Trong xUnit, mỗi testcase là một method public, được gắn nhãn [Fact]. Test Explorer sẽ tìm những hàm này và chạy và trả về kết quả.

### *iii. Mock Object*

Trong hướng đối tượng. Mock là hành động tạo ra các đối tượng, phương thức giả để giả lập hành vi của đối tượng thật. Đây là một kỹ thuật trong Unit Testing để tạo ra các Mock Object ảo không ảnh hưởng đến object thật.

### *iv. Moq*

Moq là một thư viện hỗ trợ tạo các mock object cho các ứng dụng .NET. Moq cung cấp các phương thức giả lập

## 3. Thực hiện project

Trong thời gian training với công ty, em có làm một bài practice được tổ chức trên hệ thống eLearning của công ty. Đây là bài kiểm tra đầu ra để đảm bảo rằng em có đủ kiến thức để có thể tham gia vào dự án thực tế. Nội dung công việc như sau:

- Sử dụng Docker để tạo instance SQL Server
- Sử dụng Entity Framework để tương tác với CSDL SQL Server
- Sử dụng mô hình 3 lớp để xây dựng ứng dụng ASP.NET Core API
- Tìm hiểu và vận dụng Dependency Injection, đồng thời lấy dữ liệu từ file appsetting.json để thiết lập các cài đặt.
- Vận dụng JWT Authentication và Refresh Token để xử lý xác thực người dùng
- Xử lý upload ảnh để implement chức năng thay đổi avatar cho người dùng
- Viết Unit Test cho các Business Logic Layer sử dụng xUnit và Moq
- Viết ứng dụng Frontend bằng React để đảm bảo ứng dụng API hoạt động trơn tru và chính xác.

## 4. Tham gia vào dự án công ty

### a. Tổng quan về dự án

Đây là một dự án về quản lý phòng khám và bệnh nhân, cung cấp các giải pháp về quản lý hồ sơ bệnh nhân, quản lý các đợt khám, quản lý bảo hiểm. Đây là một giải pháp toàn diện về y tế và chăm sóc sức khỏe cho các phòng khám ở Mỹ.

### b. Các công việc chính trong dự án

- Tìm hiểu về các nghiệp vụ liên quan đến y tế, bảo hiểm và chăm sóc sức khỏe. Đặc biệt là các điều khoản nghiêm ngặt về bảo mật thông tin liên quan đến y tế của Mỹ.
- Tìm hiểu và sử dụng các công cụ và môi trường được cung cấp bởi khách hàng (trong đó có Azure DevOps)
- Giao tiếp trực tiếp với khách hàng, phân tích yêu cầu về nghiệp vụ và về kỹ thuật, đồng thời truyền đạt lại những yêu cầu đó cho đội ngũ Dev
- Phát triển chức năng mới, đồng thời integrate những chức năng mà đội Dev của khách hàng đã phát triển vào codebase chính của dự án, sau đó là viết Unit Test cho những chức năng mới này để đảm bảo tính an toàn cho codebase
- Follow theo quy trình Scrum, nhận các task (ticket) từ Project Manager và giải quyết các Task trong thời gian cho phép

### 5. Lịch làm việc

Tuần	Công việc	Người hướng dẫn	Mức độ hoàn thành	Nhận xét của người hướng dẫn
1	<ul style="list-style-type: none"> <li>- Tìm hiểu về công ty, cách tổ chức của công ty.</li> <li>- Làm quen với các công cụ làm việc trong công ty.</li> <li>- Học cách trao đổi, làm việc qua email.</li> <li>- Tìm hiểu ngôn ngữ lập trình C#</li> <li>- Tìm hiểu về .NET Framework</li> <li>- Tìm hiểu về Git và cách làm việc chuyên nghiệp với Git, Git Flow</li> </ul>	Anh Trương Ngọc Duy		
2	<ul style="list-style-type: none"> <li>- Setup CSDL SQL Server với Docker</li> </ul>	Anh Trương Ngọc Duy		

	<ul style="list-style-type: none"> <li>- Tìm hiểu về tương tác với CSDL bằng ORM Entity Framework</li> <li>- Tìm hiểu về mô hình 3 lớp</li> <li>- Tìm hiểu về nguyên lý SOLID trong lập trình và áp dụng vào xây dựng ứng dụng</li> <li>- Xây dựng các API CRUD Web với ASP.NET Core API</li> <li>- Unit Test cho Service layer và Integration Test cho Data Access Layer</li> </ul>			
3	<ul style="list-style-type: none"> <li>- Tìm hiểu về JWT Authentication trong ASP.NET</li> <li>- Xây dựng xác thực với JWT Token trong ASP.NET</li> <li>- Tìm hiểu và vận dụng để xây dựng tính năng upload và lấy hình ảnh trong ASP.NET</li> </ul>	Anh Trương Ngọc Duy		
4	<ul style="list-style-type: none"> <li>- Tìm hiểu về Claim Identity và cách thức sử dụng Claims trong việc bảo mật tài nguyên người dùng</li> <li>- Tìm hiểu ReactJS và xây dựng ứng dụng ReactJS để hiển thị UI cho ứng dụng ASP.NET Core API</li> </ul>	Anh Trương Ngọc Duy		
5	<ul style="list-style-type: none"> <li>- Tổng kết, tổng hợp tài liệu, kết thúc thử việc. Ký Hợp đồng lao động</li> </ul>	Anh Trương Ngọc Duy		
6	<ul style="list-style-type: none"> <li>- Bàn giao sang dự án chính thức</li> </ul>	Anh Trương Ngọc Duy		

## Chương 3: Chi tiết về project

### 1. Giới thiệu về dự án

Đây là một giải pháp toàn diện về y tế và chăm sóc sức khỏe dành cho các phòng khám và bệnh viện. Cho đến nay sau hơn 9 năm hoạt động công ty này đã kết nối với hơn 150 triệu bệnh nhân, hơn 900 insurance payer.

Toàn bộ dự án được viết bằng ASP.NET và .NET Standard Library. Codebase của dự án sử dụng cả .NET Core và .NET Framework và do đó dự án được deploy trên Windows Server.

Vì lý do chính sách bảo mật dữ liệu nghiêm ngặt từ cả KMS lẫn công ty khách hàng nên em xin được phép không tiết lộ tên cũng như bất kỳ giao diện nào của dự án.

## TÀI LIỆU THAM KHẢO

[Cơ bản về Gitflow Workflow \(viblo.asia\)](#)

<https://topdev.vn/blog/mo-hinh-3-lop-la-gi/>

<https://comdy.vn/unit-test/tong-quan-ve-unit-test-voi-asp-net-core-xunit-va-moq/>

<https://viblo.asia/p/unit-tests-with-moq-in-net-EoDGQoBqkbV>

## TỔNG KẾT

Như vậy chỉ trong vòng hơn 1 tháng thử việc, em đã tìm hiểu rất nhiều kiến thức thực tế và trang bị cho mình đầy đủ kiến thức và kỹ năng cần thiết để có thể làm việc cho một dự án chính thức. Do thời gian và chính sách bảo mật nên em chỉ có thể hoàn thành dự án luyện tập dành cho Fresher của nội bộ công ty và em không thể giới thiệu chi tiết về dự án này cũng như dự án mà hiện tại em đang làm. Tuy nhiên khi có thời gian em sẽ vận dụng những kiến thức này để hoàn thành một dự án khác có nhiều chức năng và hoàn thiện hơn.

Chân thành cảm ơn sự giúp đỡ nhiệt tình của anh Trương Ngọc Duy, LnD và team Admin của KMS đã hỗ trợ và hướng dẫn em để em có thể hoàn thành bài tập cũng như tham gia vào dự án chính thức.