# TEDU-53: Xây dựng CMS với .NET 8 & Angular

# Nội dung chính khóa học

## Chương 1: Giới thiệu và chuẩn bị cho dự án

### 1. Giới thiệu tổng quan khóa học

Khóa học xây dựng dự án thực tế sử dụng phiên bản .NET mới nhất sẽ ra mắt chính thức vào cuối năm 2023 là .NET 8. Khóa học này [Clean Architecture by Uncle Bob](https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html) sẽ cung cấp cho các bạn những kiến thức và cách xây dựng dự án được cập nhật công nghệ mới nhất. Song song với đó, nó cũng cung cấp các best practices trong quá trình làm việc của giảng viên.

Một số best practice phải kể đến như:

1. Clean Architecture: <https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html>
2. Distributed Architecture: <https://www.geeksforgeeks.org/architecture-styles-in-distributed-systems/>
3. API Client generate: <https://learn.microsoft.com/en-us/aspnet/core/tutorials/getting-started-with-nswag?view=aspnetcore-7.0&tabs=visual-studio>
4. Domain Driven Design: <https://learn.microsoft.com/en-us/archive/msdn-magazine/2009/february/best-practice-an-introduction-to-domain-driven-design>

Khóa học này sử dụng Angular làm hệ quản trị cho administrator, đây là một trong những frontend framework phổ biến nhất giúp bạn có thêm cả kỹ năng làm việc với frontend, để hướng tới trở thành full stack developer.

Phần website, sẽ sử dụng công nghệ ASP.NET Core MVC để giúp các bạn củng cố thêm kiến thức về mô hình MVC, bổ trợ cho phần REST API trong phần quản trị. Đây sẽ là kỹ năng xuyên suốt toàn bộ khóa học sát với thực tế nhất. Khi mà các dự án trong thực tế gần như 100% sử dụng mô hình REST API và MVC.

Cụ thể ứng dụng chúng ta sẽ xây dựng là một CMS để giúp chúng ta tạo một blog hiển thị bài viết và quản trị website.

Các tính năng chính:

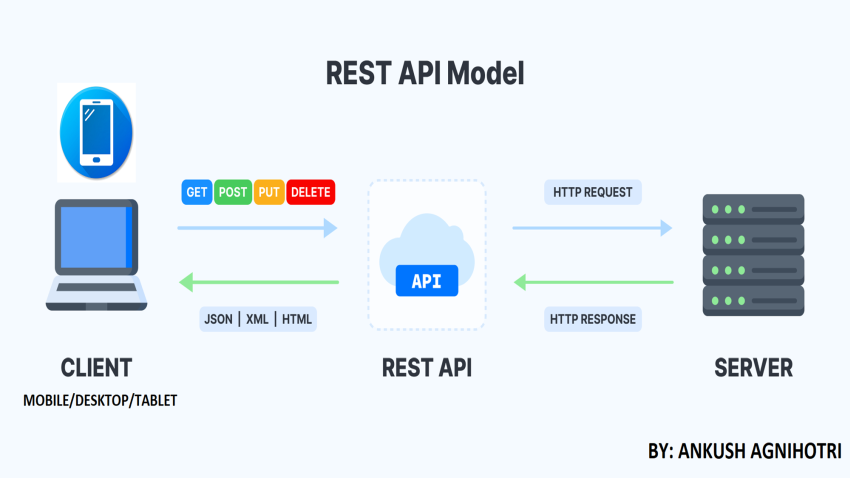
* Admin: Phân quyền, quản trị bài viết, danh mục, loạt bài, tác giả và nhuận bút, duyệt bài, đăng bài.
* Portal: Hiển thị tin tức, tag, tìm kiếm, đăng ký đăng nhập, đăng bài, xem thống kê.

### 2. Giới thiệu mô hình MVC

<https://learn.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-8.0&WT.mc_id=dotnet-35129-website>

### 3. Tìm hiểu về REST và RESTful API

Tài liệu: <https://restfulapi.net/>



### 4. Tổng quan về nền tảng .NET

Giới thiệu về .NET: <https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet>

.NET Roadmap: <https://dotnet.microsoft.com/en-us/platform/support/policy/dotnet-core>

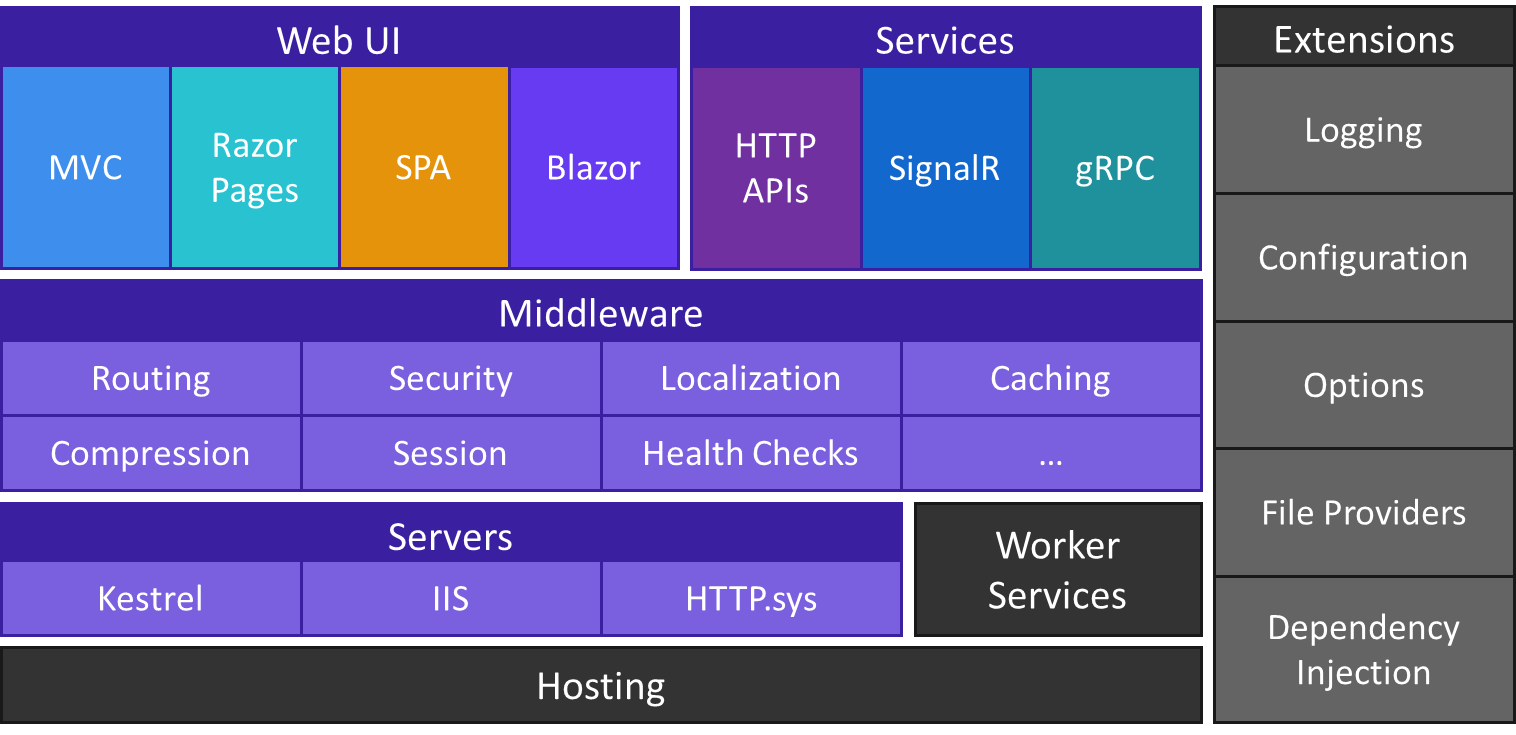
Lộ trình .NET Developer 2023: <https://github.com/MoienTajik/AspNetCore-Developer-Roadmap/blob/master/aspnetcore-developer-roadmap.png>

### 5. Tìm hiểu về ASP.NET Core

<https://learn.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-7.0>

<https://learn.microsoft.com/en-us/aspnet/core/getting-started/?view=aspnetcore-8.0&tabs=windows>

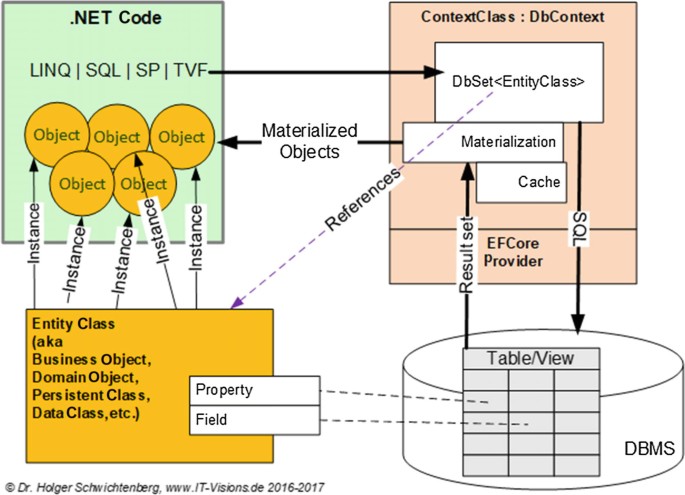
Tham khảo: <https://code-maze.com/asp-net-core-mvc-series/>



### 6. Giới thiệu về Entity Framework Core

Tài liệu: <https://learn.microsoft.com/en-us/ef/core/>

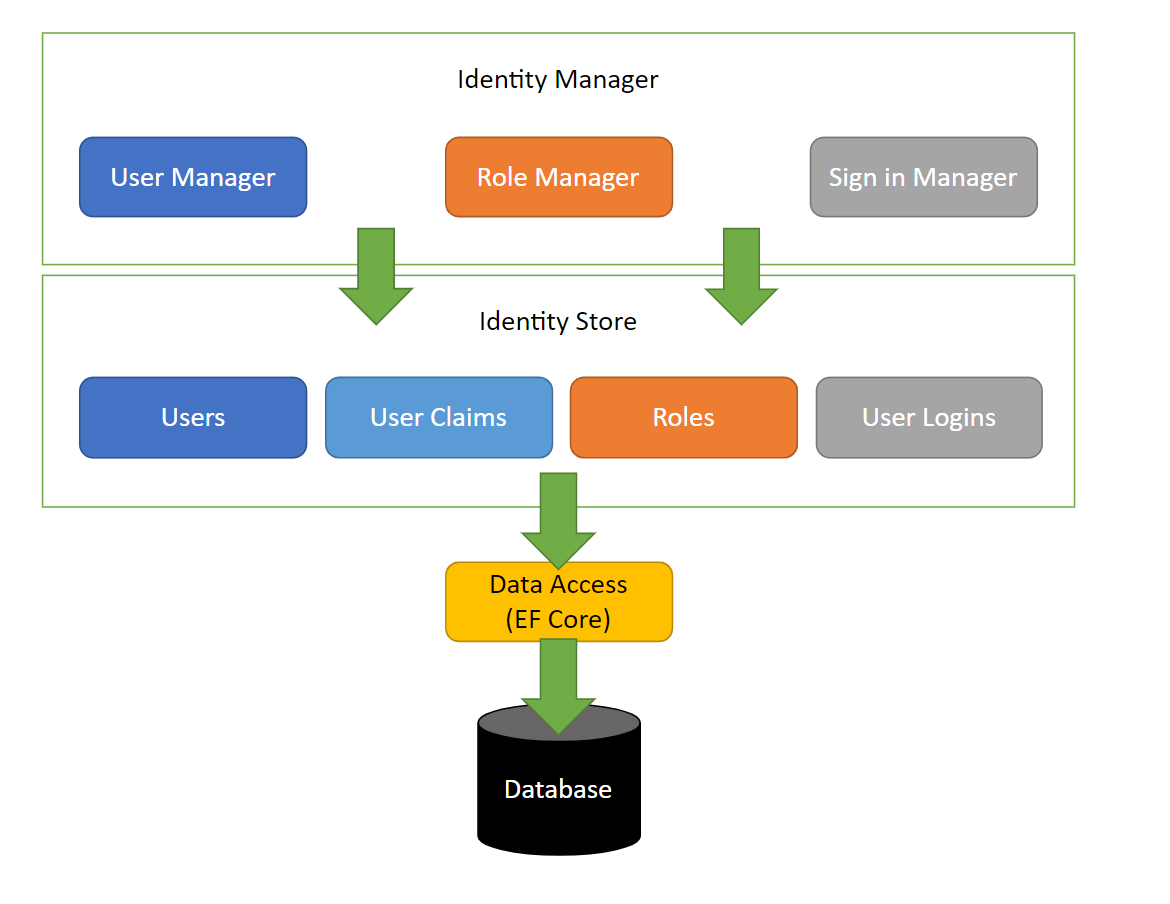
Tham khảo: <https://code-maze.com/entity-framework-core-series/>



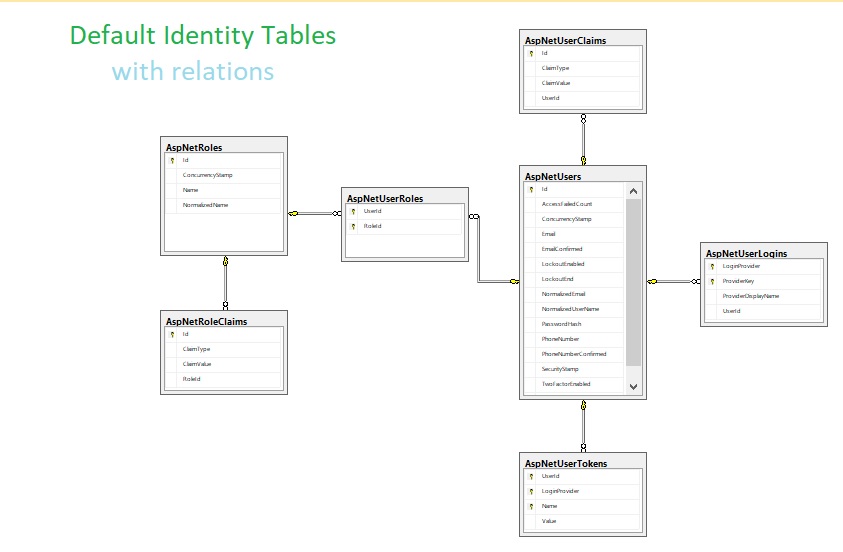
### 7. Giới thiệu về ASP.NET Identity

Tài liệu: <https://learn.microsoft.com/en-us/aspnet/core/security/authentication/identity?view=aspnetcore-7.0&tabs=visual-studio>

Tham khảo: <https://code-maze.com/asp-net-core-identity-series/>

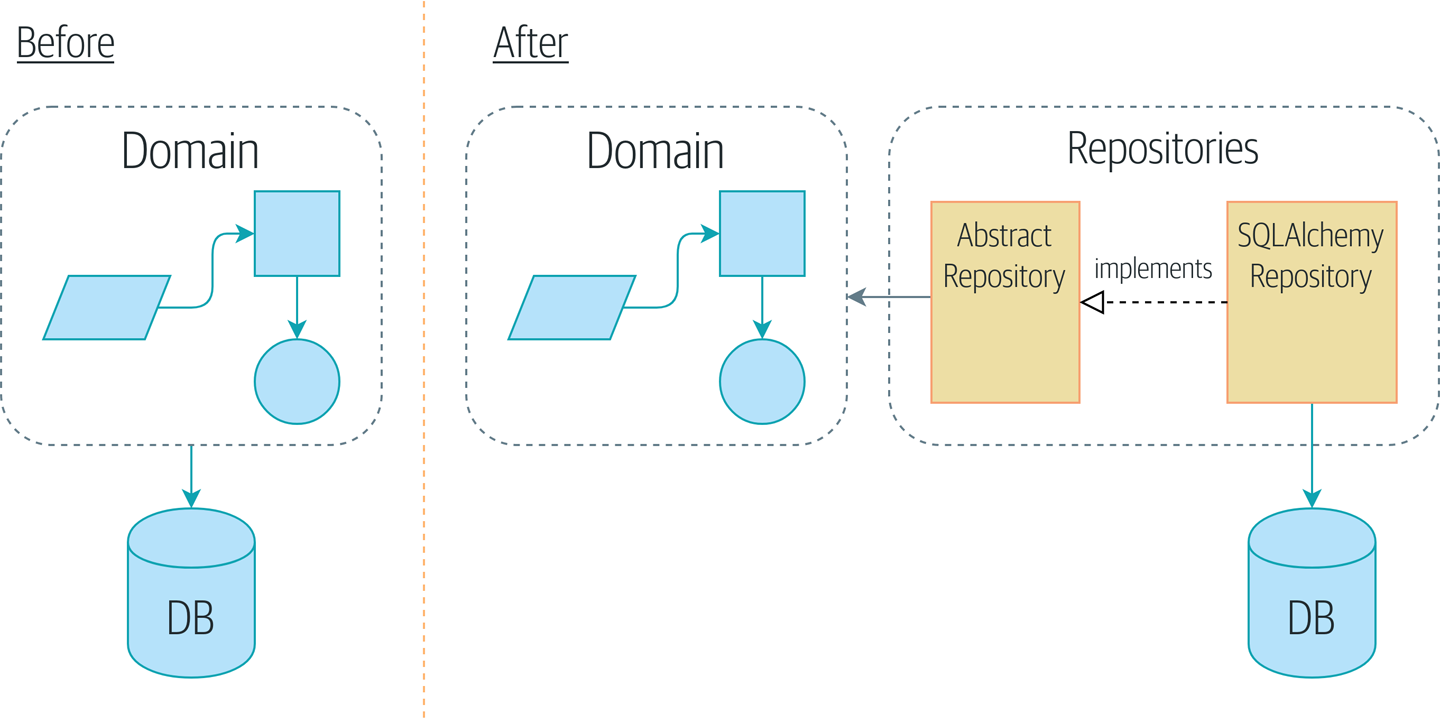


Bảng mặc định

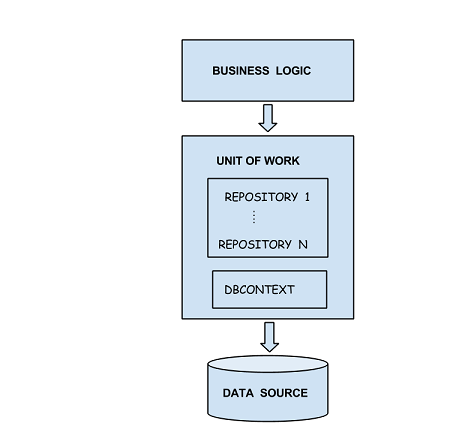


### 8. Giới thiệu một số Design Pattern sử dụng trong khóa học

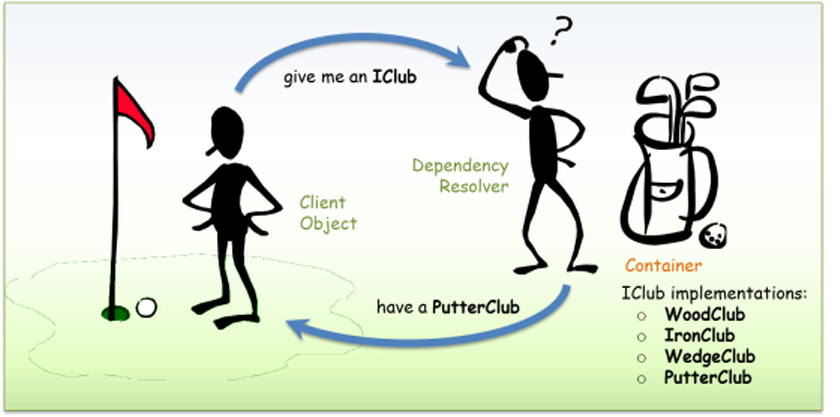
1. Repository



1. Unit Of Work



1. Dependency Injection

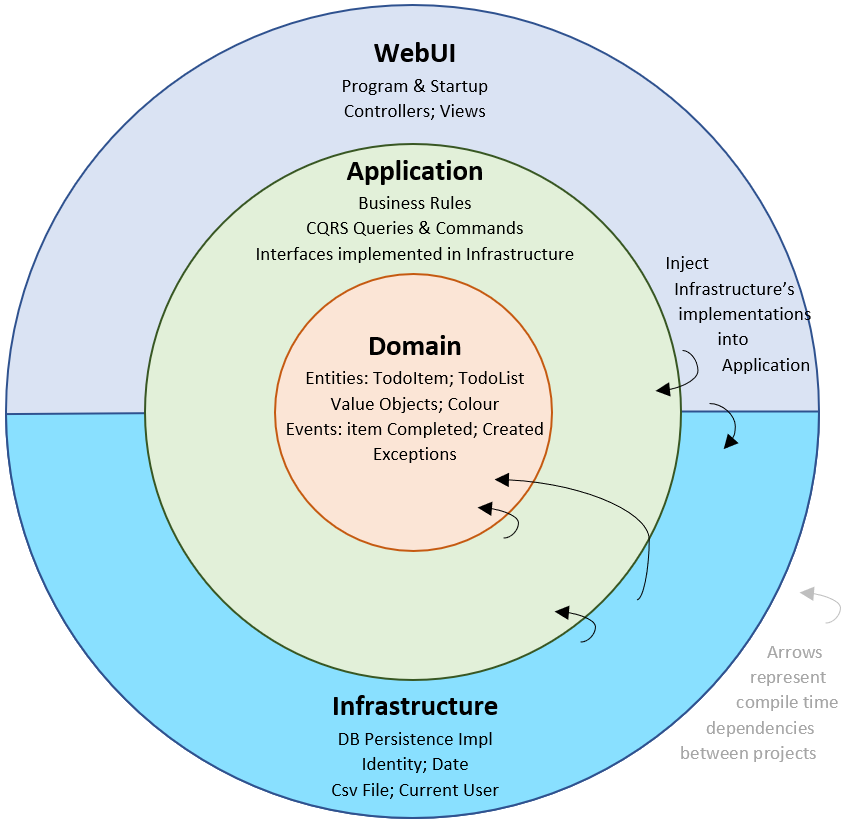


1. DDD

Tham khảo:

* <https://viblo.asia/p/domain-driven-design-phan-1-mrDGMOExkzL>
* <https://tigosoftware.com/vi/mo-hinh-hoa-huong-nghiep-vu-domain-driven-design-ddd-la-gi>
* <https://cloudgeeks.net/tim-hieu-ve-ddd-domain-driven-design-moi-quan-he-giua-ddd-va-microservice/>
* <https://learn.microsoft.com/en-us/archive/msdn-magazine/2009/february/best-practice-an-introduction-to-domain-driven-design>

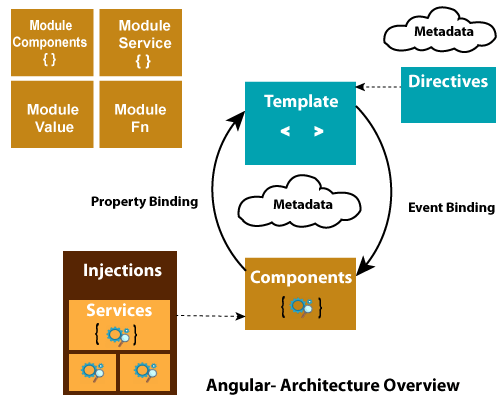
1. Clean Architect

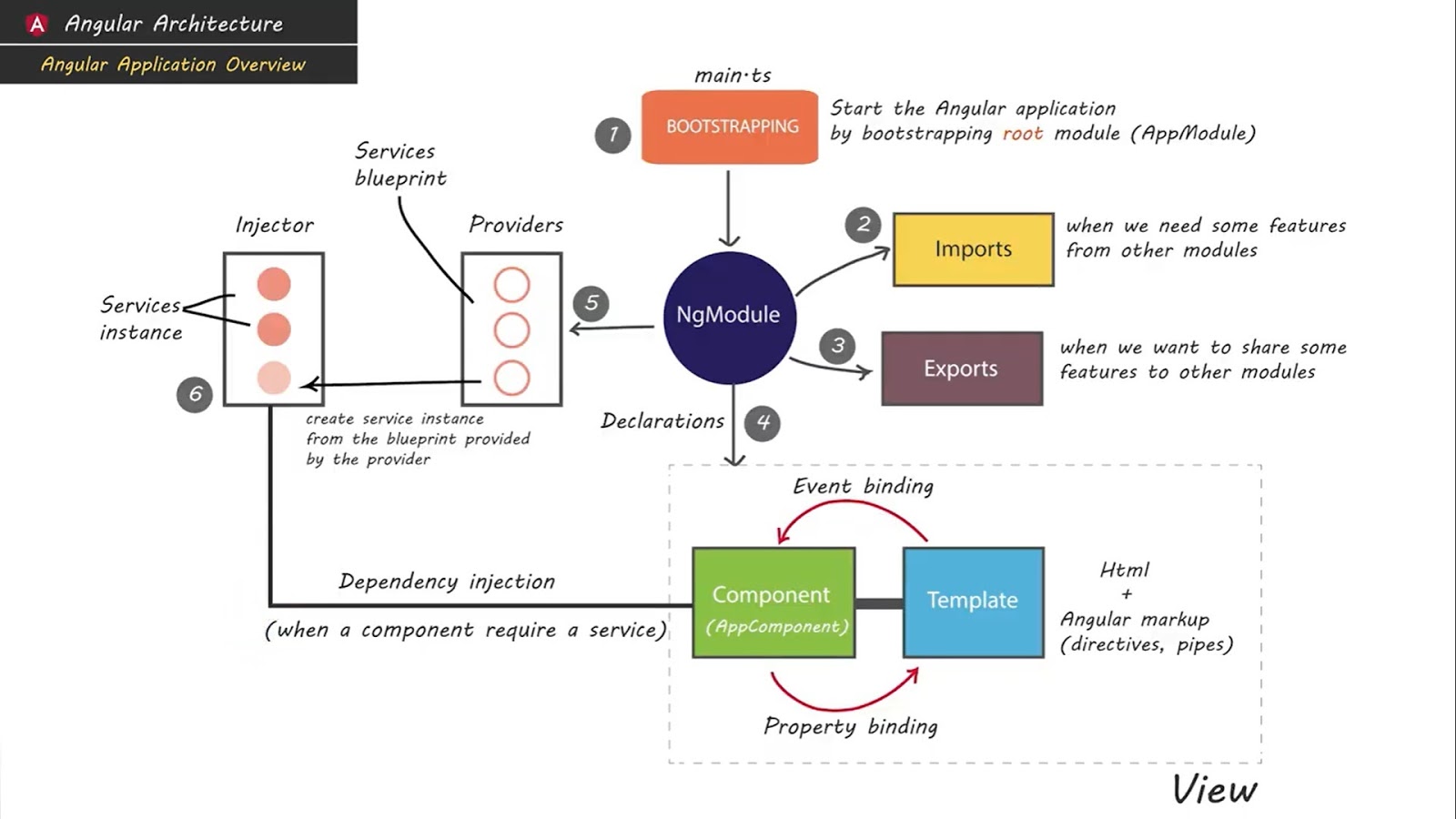


### 9. Tìm hiểu về Angular Framework

Trang chủ: <https://angular.io/guide/what-is-angular>

Tổng quan: <https://angular.io/guide/architecture>

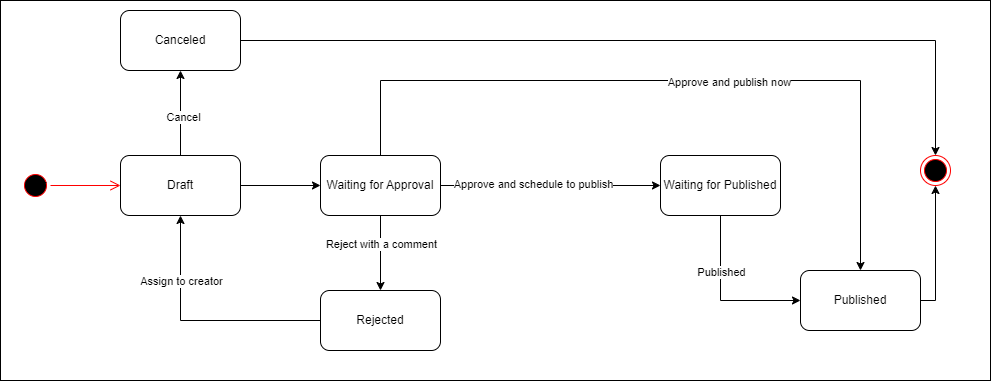




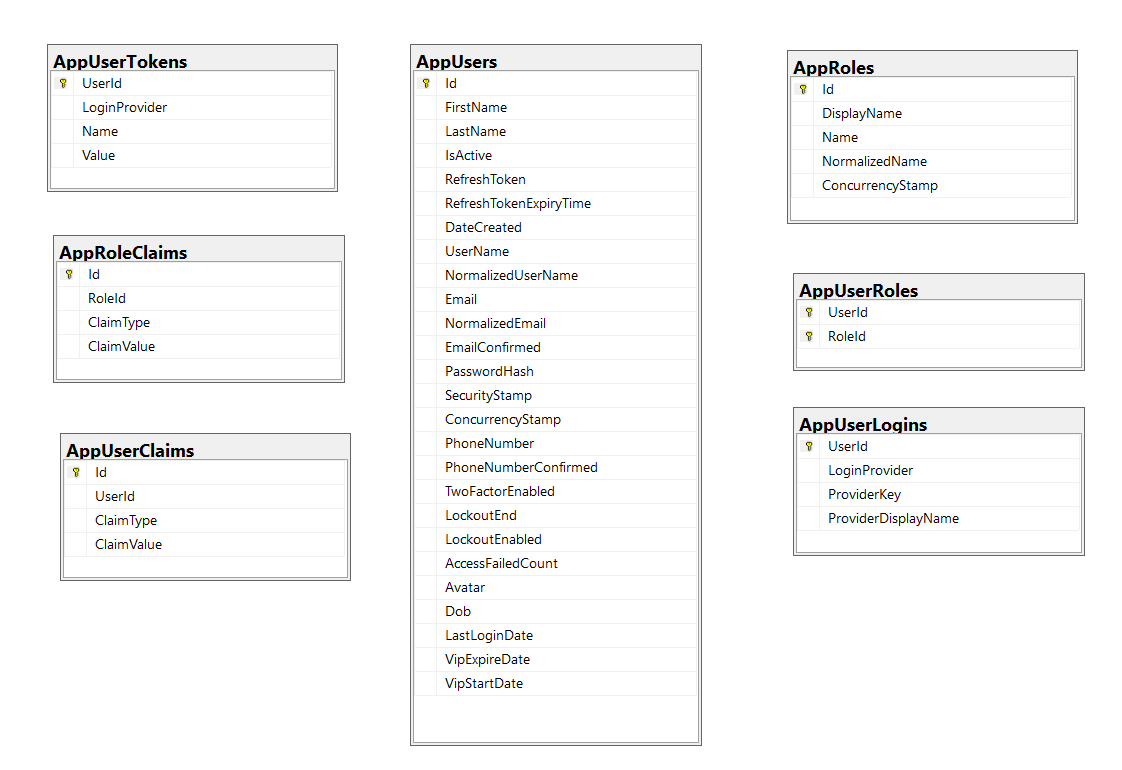
### 10. Giới thiệu về dự án CMS

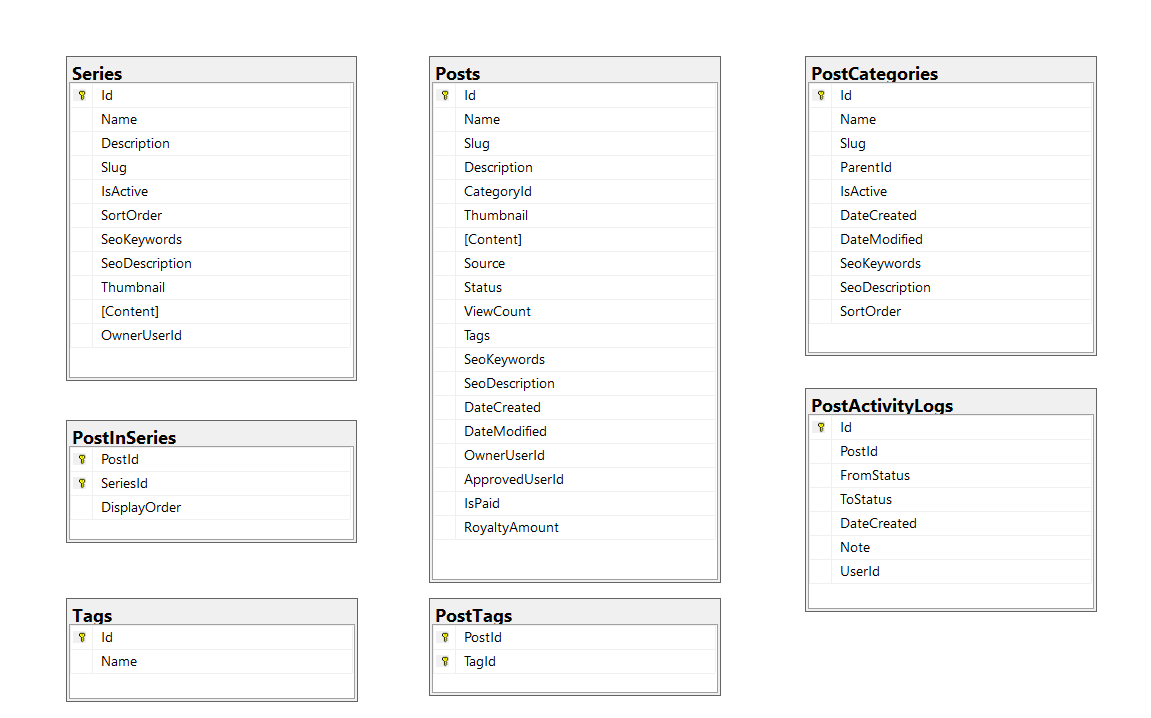
* Trang quản trị
  + Quản lý role
  + Quản lý user (gán quyền, phân quyền, đổi email, mật khẩu)
  + Quản lý danh mục
  + Quản lý tin (duyệt bài, đăng bài)
  + Quản lý loạt bài
  + Quản lý comment
* Trang tin tức
  + Hiển thị trang chủ
  + Danh sách tin
  + Chi tiết tin
  + Đăng ký, đăng nhập
  + Đăng nhập qua Google, Facebook
  + Xem danh sách tin đã đăng
  + Loạt bài đã đăng
  + Thông tin nhuận bút
  + Thông báo

### 11. Tìm hiểu về nghiệp vụ dự án



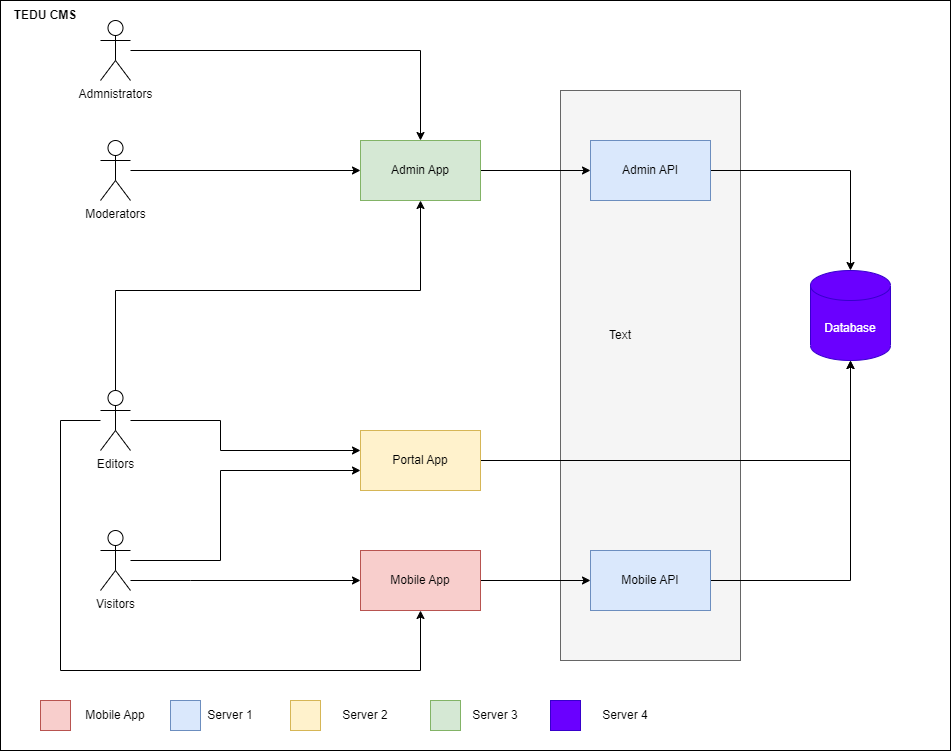
### 12. Phân tích thiết kế cơ sở dữ liệu





## Chương 2: Xây dựng hạ tầng cho dự án

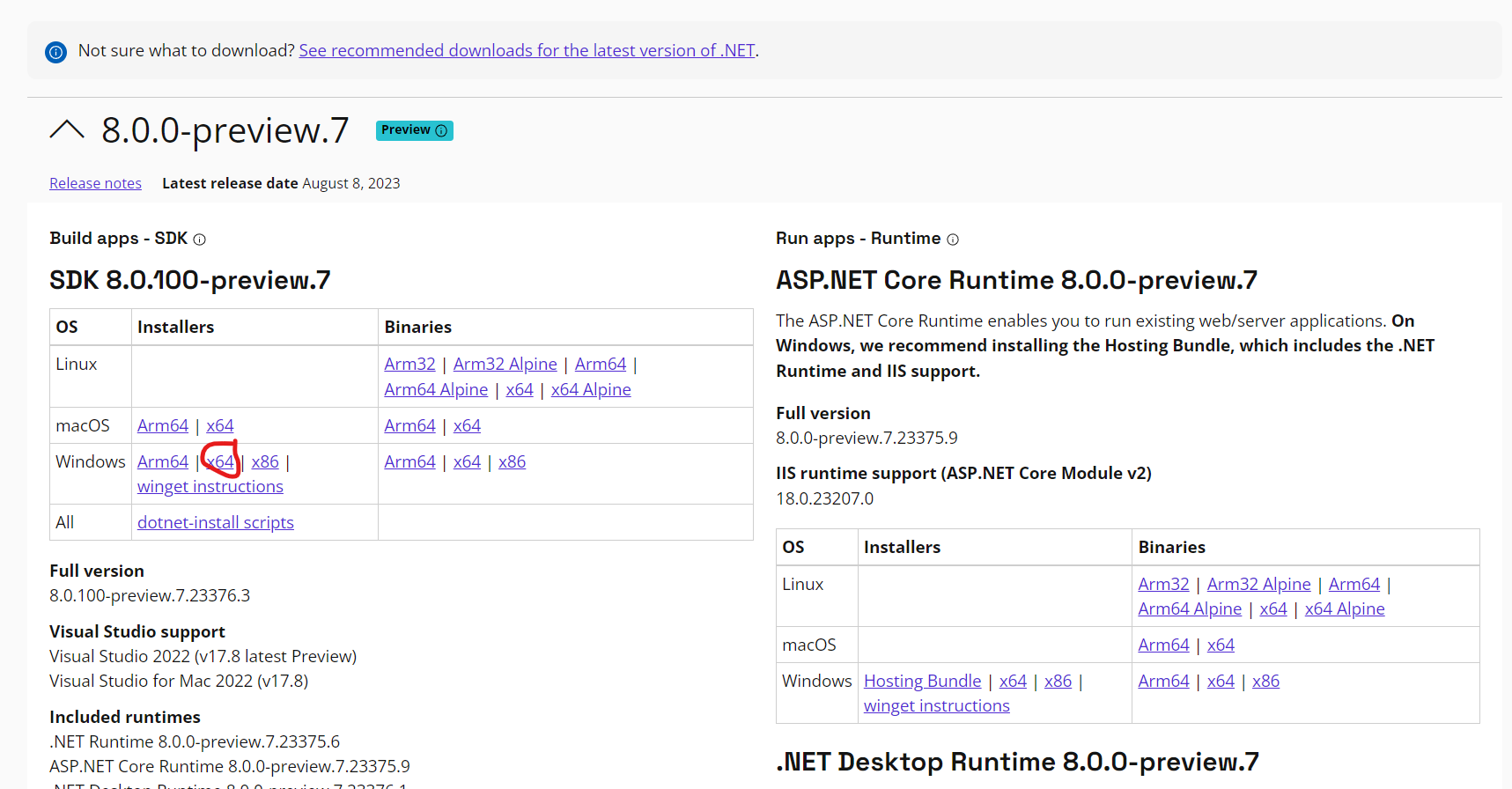
### 1. Giới thiệu mô hình deploy ứng dụng



### 2. Cài đặt môi trường phát triển

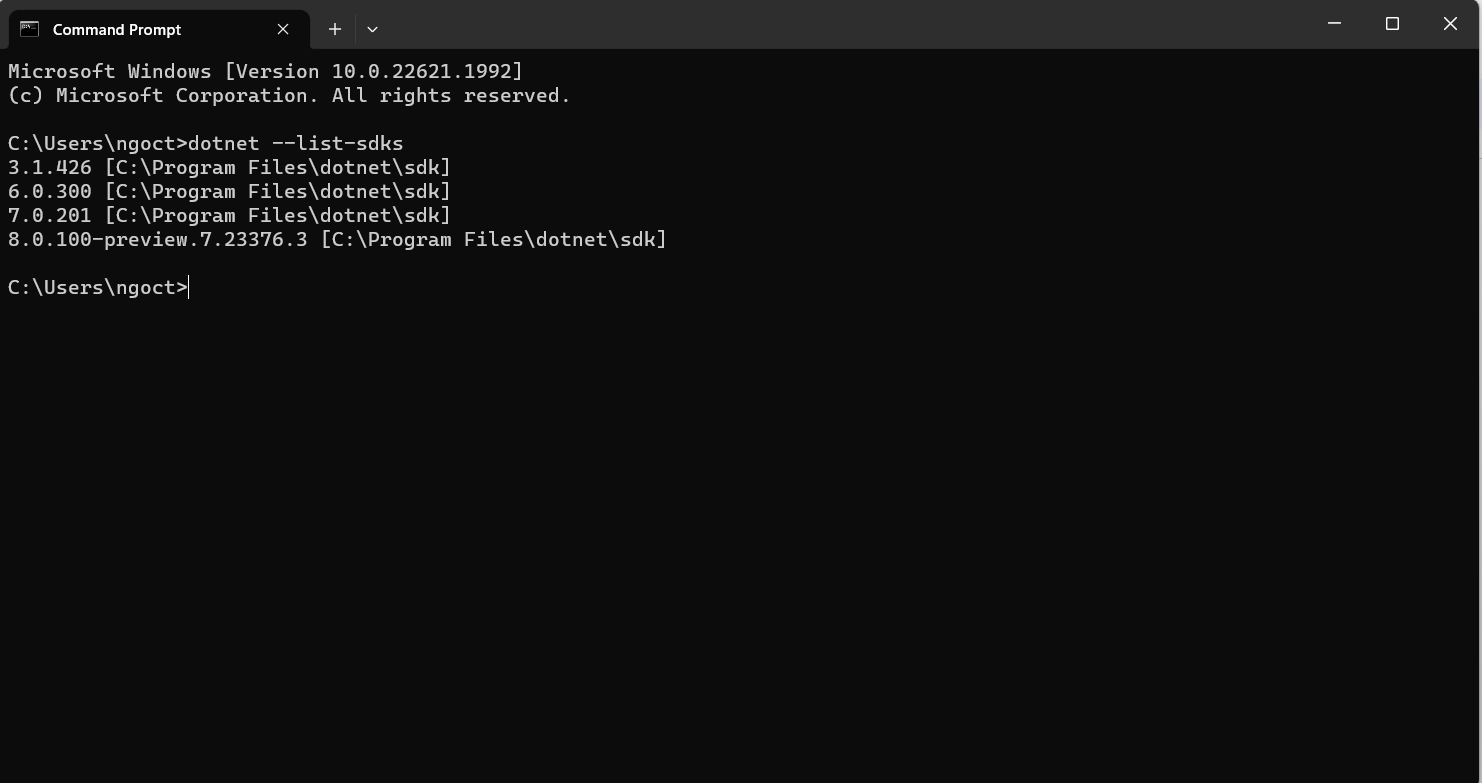
Đối với các bạn cài đặt mới thì cần cài những tool sau:

* .NET 8 SDK: [Download .NET 8.0 (Linux, macOS, and Windows)](https://dotnet.microsoft.com/en-us/download/dotnet/8.0) (Hiện đang là bản Preview, sẽ ra bản chính thức vào tháng 11/2023 nhưng các bạn yên tâm lúc đó upgrade dễ lắm)



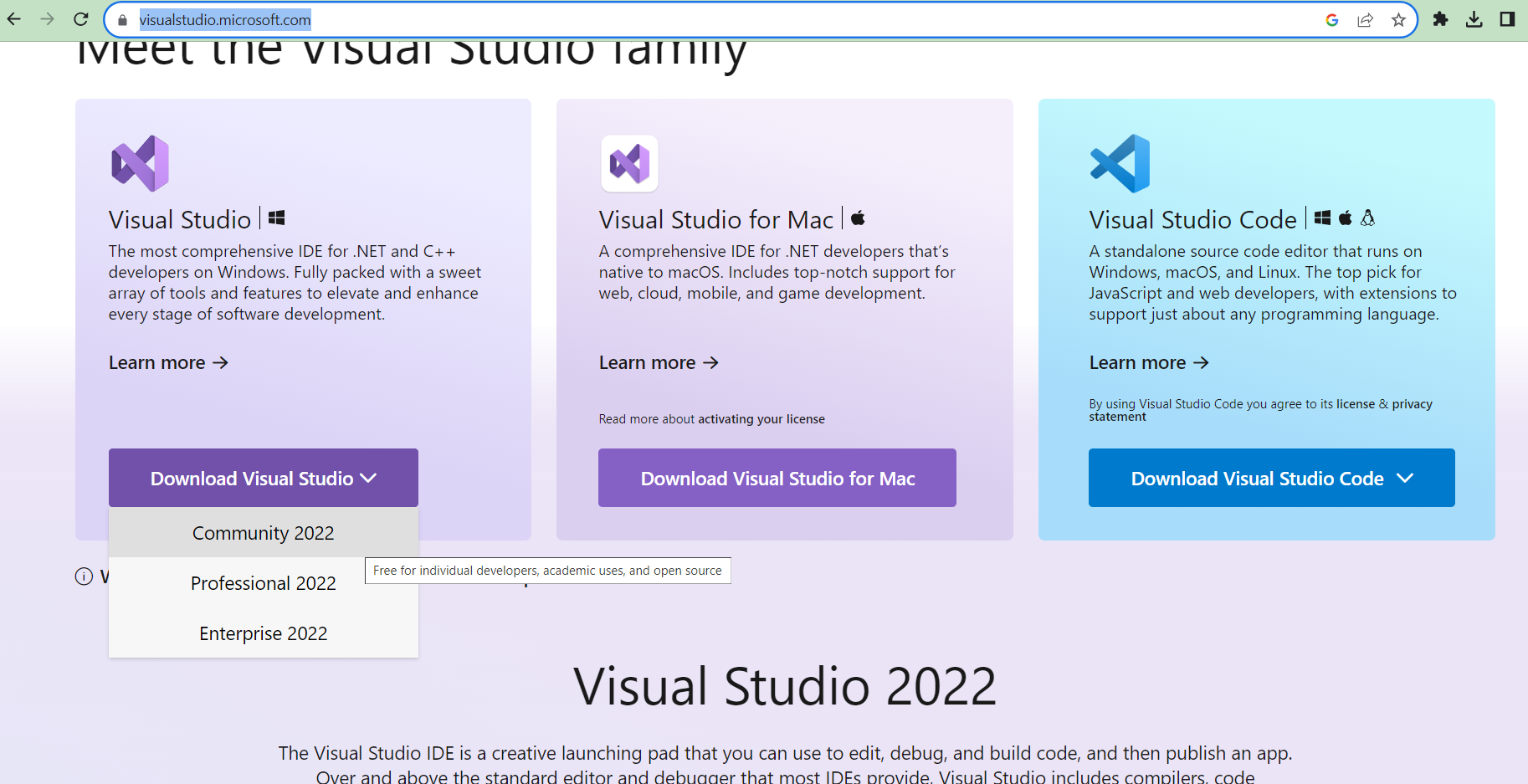
Các bạn chọn cột trái là Installer rồi chọn x64, vì đa phần máy tính Windows là nhân 64bit rồi nên chúng ta chọn x64.

Sau khi cài xong các bạn có thể kiểm tra danh sách các SDK cài trên máy bằng câu lệnh: **dotnet --list-sdks** trên cửa sổ **Command Prompt** của Windows

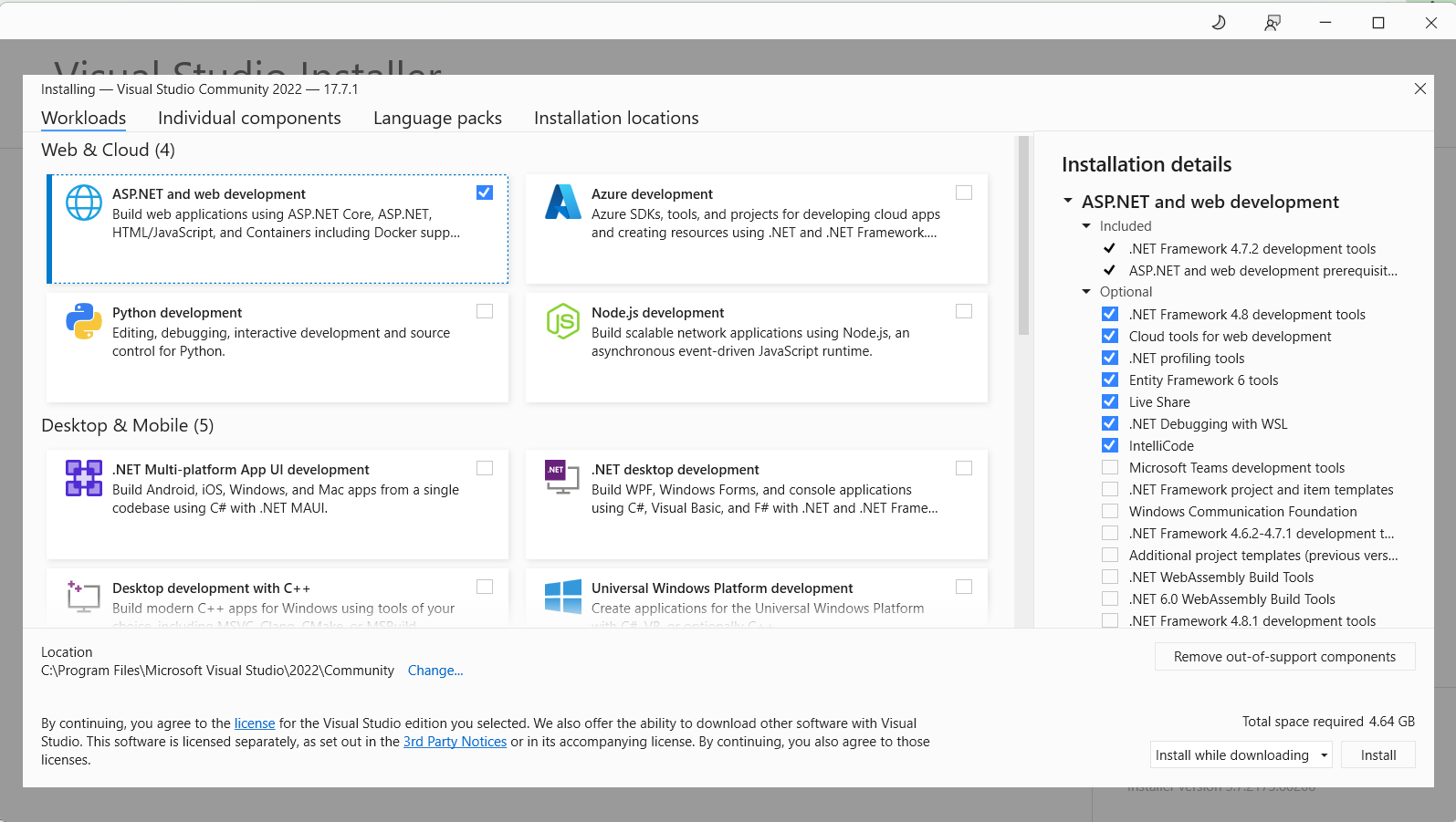


Như vậy là chúng ta đã cài thành công, các bạn nhìn thấy máy mình đang cài 1 số phiên bản SDK của .NET từ 3.1 cho đến 8.0. Khóa học này chúng ta sẽ dùng bản 8.0.

* Visual Studio 2022 tại: <https://visualstudio.microsoft.com/>. Các bạn có thể cài bản Professional hay bản Community đều được.

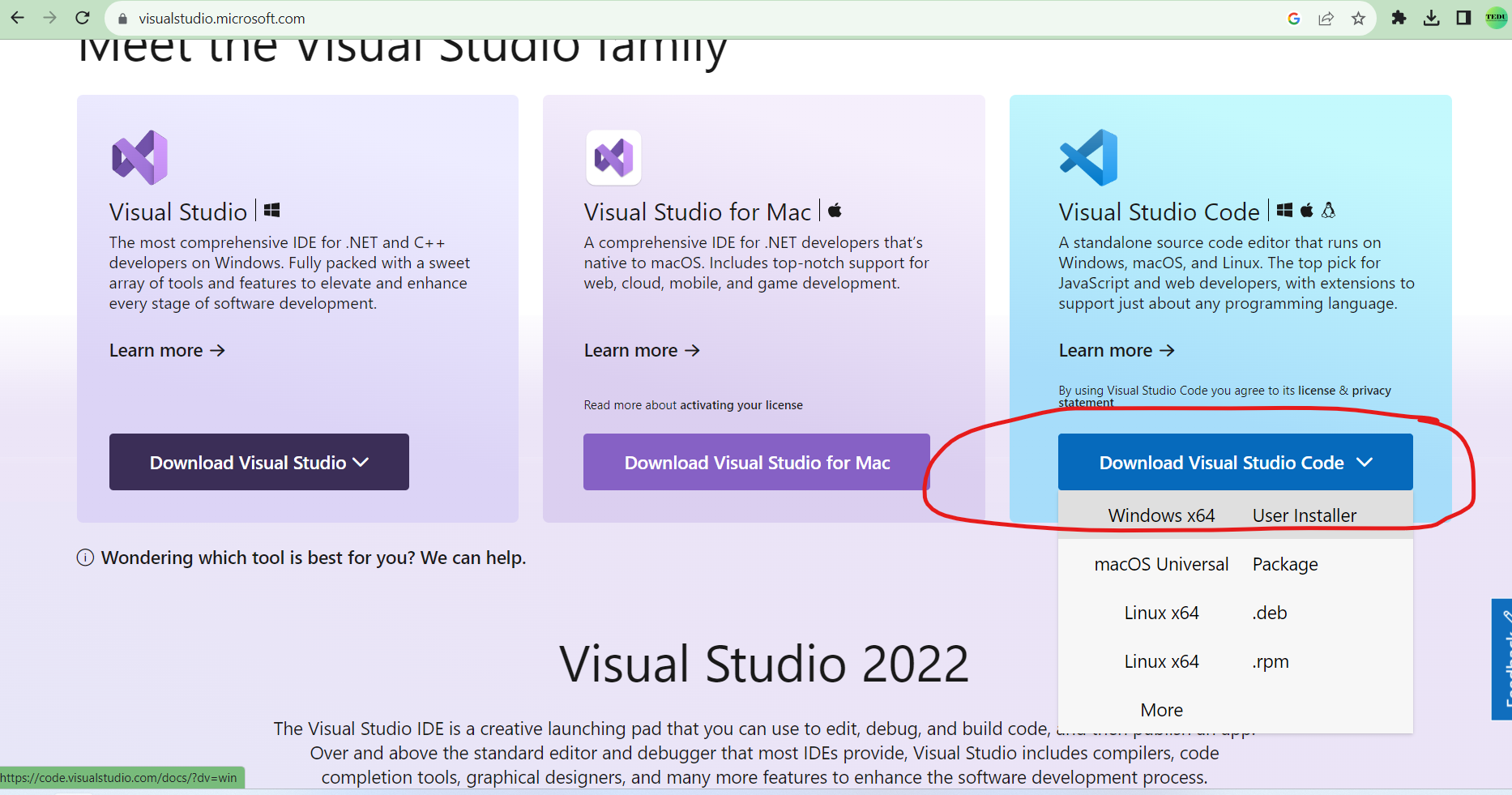


Sau khi download các bạn click vào file cài đặt rồi mở lên bấm Continue, lên cửa sổ cài đặt các bạn chỉ cần chọn **ASP.NET and web development** là ok nhé:



Sau đó bấm **Install** và chờ kết quả thôi.

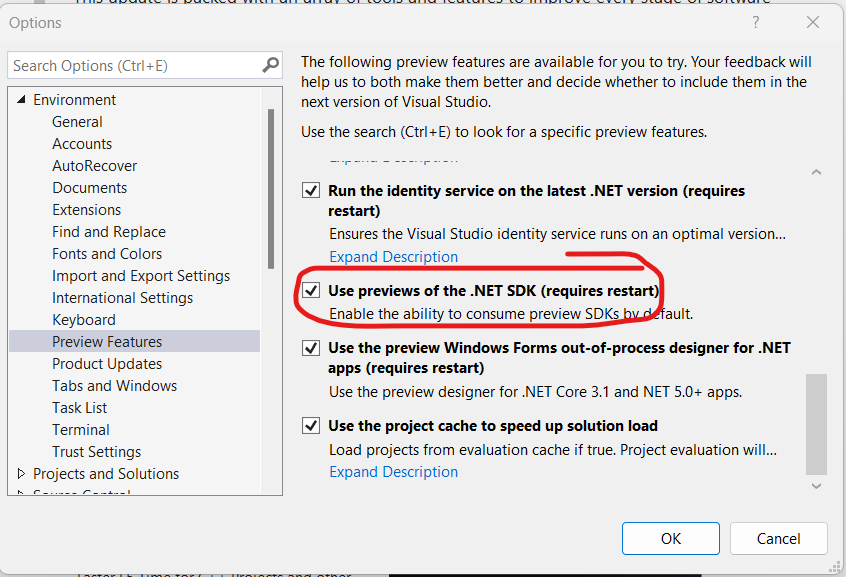
* Visual Studio Code: Để code Angular các bạn cần cài Visual Studio Code, cũng ở đường link trên



* Tiếp theo là cài đặt NodeJS: <https://nodejs.org/en> để chạy code Frontend. Các bạn cứ dùng bản nào mà Recommended for Most Users là ok. Cài xong các bạn mở cửa sổ CMD gõ **node -v** mà ok là được.
* Tiếp theo cài đặt Git: <https://git-scm.com/> đây là một source control management nổi tiếng và phổ biến nhất bây giờ. Cài xong các bạn vào cửa sổ CMD gõ **git** và **Enter** mà nó không lỗi là ok.
* Cài đặt SQL Server: <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>. Cài đặt bản Developer.
* SQL Server Management Studio: <https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16>

Đối với các bạn đã có sẵn Visual Studio thì các bạn cần:

* Update Visual Studio 2022 lên bản mới nhất: Vào menu Help → chọn **Check for Updates → Update**
* Do .NET 8 hiện tại đang là bản Preview 7, nên chúng ta không thể tạo ứng dụng với nó luôn nên phải bật chế độ cho phép dùng bản Preview. Để bật chế độ này lên các bạn vào: Trên Visual Studio 2022 chọn Tools → Options



Sau đó chọn OK và restart Visual Studio là được.

### 3. Tạo solution dự án từ Scratch

* Tạo blank solution
* Tạo 3 project: Web API, Core, Data

### 4. Kết nối và gửi source code lên GitHub

* Tại thư mục của solution: Gõ lệnh git init
* Gõ lệnh git commit -m “Initial”
* Add remote repository sau khi tạo

### 5. Tạo các entity cho phần bài viết

### 6. Tích hợp ASP.NET Core Identity

### 7. Config migrate database

### 8. Seeding dữ liệu vào database khi khởi động

### 9. Cấu hình Repository và UnitOfWork

### 10. Thử nghiệm repository cho Post

### 11. Tạo base phân trang và AutoMapper

### 12. Test thử Controller cho Post

## Chương 3: Xây dựng phần Back Office

### 1. Tạo ứng dụng Angular CLI với CoreUI

Link coreui: <https://coreui.io/angular/>

Cài đặt Angular CLI: <https://angular.io/guide/setup-local>

Cài đặt Yarn: <https://classic.yarnpkg.com/lang/en/docs/install/#windows-stable>

### 2. Xây dựng cấu trúc module và cấu hình Routing

Sẽ có 4 module chính:

1. Auth
2. System
3. Content
4. Dashboard

### 3. Hoàn thiện menu và chỉnh lại khung template

### 4. Authentication & Authorization với API JWT Token - Phần 1

* Đăng nhập và generate token

### 5. Authentication & Authorization với API JWT Token - Phần 2

* Get quyền ra theo role

### 6. Cấu hình generate API Client cho Angular với NSwag

* [**https://github.com/RicoSuter/NSwag**](https://github.com/RicoSuter/NSwag)

### 7. Authentication & Authorization với API JWT Token - Phần 3

* Xây dựng trang login trên Angular

### 8. Hoàn thiện phần đăng nhập trên Angular

* Bài này chúng ta sẽ tạo token service để lưu và sử dụng token trong Angular

### 9. Thêm loading vào login và logout

* Destroy observable
* Thêm loading
* Logout

### 10. Phân quyền phía Angular

* Phân quyền sử dụng <https://angular.io/api/router/CanActivate>
* Lấy danh sách của các permissions được trả về từ Backend
* Ẩn hiện menu dựa trên quyền
* Tạo 1 guard cho phép hoặc không cho phép truy cập 1 router dựa trên quyền của user đó.

### 11. Sử dụng access token và refresh token

* Gán token lên API Request sử dụng Interceptor
* Refresh token tham khảo: <https://code-maze.com/using-refresh-tokens-in-asp-net-core-authentication/>

### 12. Quản lý quyền - Phần 1

* Backend API

### 13. Quản lý quyền - Phần 2

* Frontend danh sách quyền

### 14. Quản lý quyền - Phần 4

* Thêm sửa xóa quyền

### 15. Quản lý quyền - Phần 5

* Phân quyền cho từng role

### 16. Quản lý người dùng

* Danh sách người dùng
* Thêm sửa xóa người dùng
* Gán user vào role
* Đổi email
* Set mật khẩu

### 17. Quản lý danh mục bài viết

* Tạo API Controller
* Tạo component mới, include vào content module
* Thêm routing
* thêm permission
* Generate client api
* Table: PostCategories
* Entity: PostCategory

### 18. Quản lý bài viết và series kèm quy trình duyệt

* Quản lý bài viết (kèm editor và đăng ảnh)
  + Xem lịch sử thay đổi trạng thái
  + Thêm bài viết vào series
* Quản lý loạt bài viết (series)
  + Danh sách bài viết thuộc series
* Quy trình duyệt bài viết: Chia ra làm người đăng và người duyệt, người có quyền đăng bài thì chỉ tạo bài viết và lưu nháp, sau đó đẩy kiểm duyệt.
* Người kiểm duyệt sẽ có quyền Approve và được đăng bài hoặc trả lại với 1 comment.
* Các bài ở trạng thái nháp sẽ chỉ người tạo mới thấy

### 19. Quản lý nhuận bút

Technology

* Sử dụng Dapper để dùng Raw SQL truy vấn database

Backend

* Thêm trường Balance vào user để ghi nhận số dư của user
* Thêm trường LoyaltyAmountPerPost vào User để ghi nhận số tiền nhuận bút của user đó khi bài được đăng
* Thêm trường PaidDate vào Posts để ghi nhận ngày thanh toán
* Thêm bảng Transactions để ghi nhận lịch sử giao dịch
* Thêm LoyaltyService để làm các công việc get report và thanh toán

Frontend:

* Thêm màn hình lấy ra nhuận bút theo từng user, tìm kiếm theo thời gian và username
* Thêm màn hình lấy ra nhuận bút theo từng tháng của tất cả user, có tìm kiếm
* Thêm màn hình lịch sử giao dịch
* Thêm trường số tiền trên bài viết cho form tạo mới/ cập nhật user
* Thêm trường Thanh toán trên danh sách bài viết

### 20. Fix một số bugs tồn tại

* Không cho xóa series, post category khi chứa bài viết
* Xóa role từ user bị lỗi
* Tạo user mới không đăng nhập được
* Phân trang các màn hình list
* Lỗi xóa category
* Tạo loạt bài mà không check vào checkbo Kích hoạt thì báo lỗi

Tham khảo:

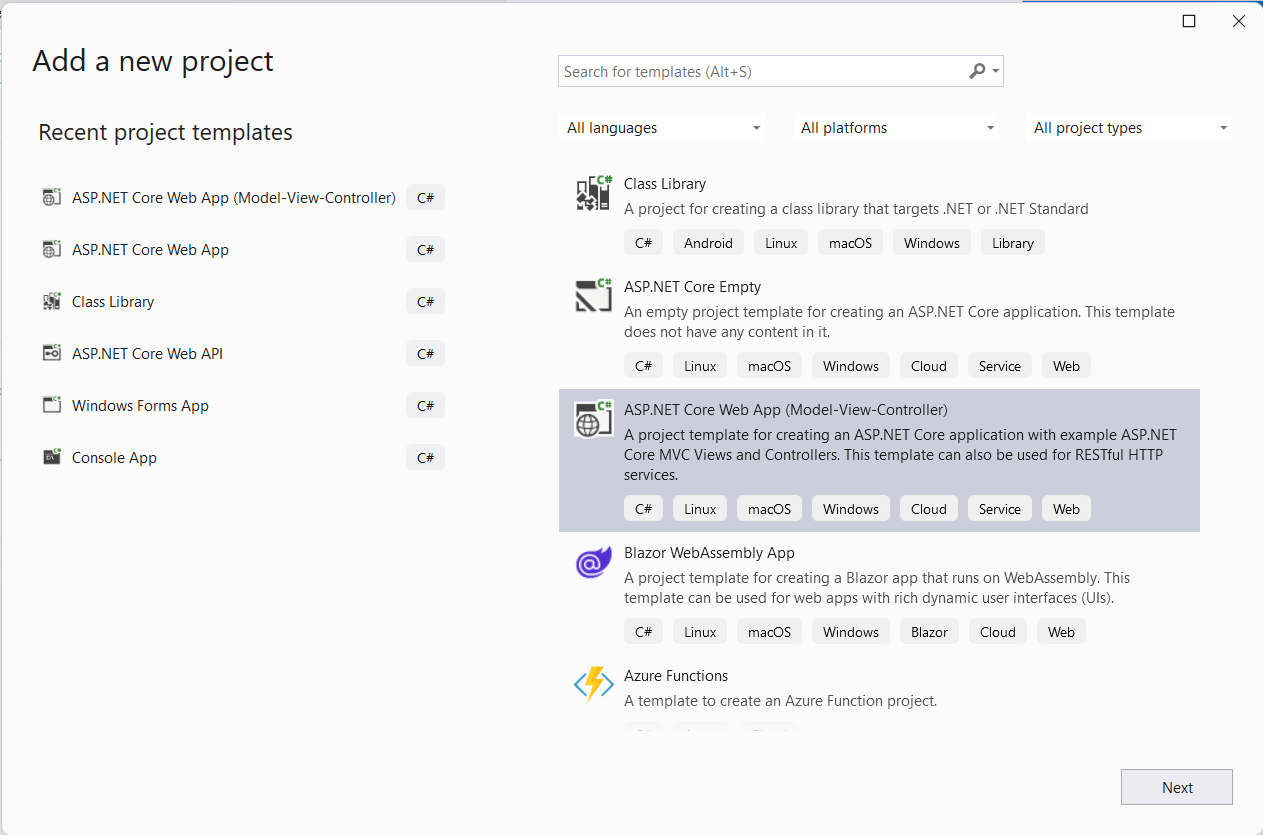
Pull request: <https://github.com/teduinternational/tedu-blog/pull/2/files>

Fix bug refresh token: <https://github.com/teduinternational/tedu-blog/pull/3>

## Chương 4: Xây dựng phần Web cho người dùng

### 1. Ghép giao diện HTML (Clean Blog)

* Giao diện: <https://startbootstrap.com/theme/clean-blog>
* Tạo ứng dụng ASP.NET Core Web App (Model - View - Controller)



### 2. Xây dựng các thư viện dùng chung

* Sử dụng các thành phần dùng chung như ParitalView và ViewComponent trong ASP.NET Core
* Cấu hình kết nối database sử dụng Entity Framework Core
* Cấu hình System config
* Cấu hình User Claim Factory
* Cấu hình sử dụng nhiều file appsettings.json theo từng môi trường.

### 3. Nâng cấp hệ thống lên .NET 8.0 (bản chính thức)

* Cài đặt .NET 8.0 SDK: <https://dotnet.microsoft.com/en-us/download/dotnet/8.0>
* Upgrade Visual Studio lên 17.8 → Helps → Check Updates
* Upgrade các thư viện liên quan lên 8.0
* Upgrade nswag lên Net80
* Thử chạy lại hệ thống
* Tham khảo: <https://learn.microsoft.com/en-us/aspnet/core/migration/70-80?view=aspnetcore-8.0&tabs=visual-studio>

### 4. Xây dựng trang chủ

* Tạo một số share view sử dụng Partial View, ViewComponent
* Bind dữ liệu trang chủ

### 5. Hiển thị danh sách bài viết theo danh mục

* Hiển thị danh sách bài viết phân trang theo danh mục
* Tạo ViewComponent phân trang danh mục
* Tối ưu SEO

### 6. Hiển thị chi tiết bài viết

* Hiển thị chi tiết bài viết

### 7. Hiển thị bài viết theo tag - Part 1

* Quản lý tag ở phía Admin

### 8. Hiển thị bài viết theo tag - Part 2

* Hiển thị danh sách tag ở chi tiết bài viết
* Danh sách bài viết theo tag

### 9. Danh sách loạt bài

* Fix bug thêm thứ tự trong loạt bài ở admin
* Hiển thị danh sách loạt bài
* Hiển thị danh sách bài viết trong từng loạt bài

### 10. Đăng ký thành viên

* Tạo form đăng ký user mới
* Đăng ký thành viên mới
* Raise event sử dụng MediatR

### 11. Đăng nhập

* Đăng nhập và gán Claim sử dụng ASP.NET Core Identity
* Viết Extension methods để lấy thông tin claims

### 12. Xem và cập nhật profile

* Xem profile
* Cập nhật profile
* Đăng xuất

### 13. Đổi mật khẩu, quên mật khẩu

* Tạo form đổi mật khẩu
* Tạo form quên mật khẩu → Gửi mail → Tích hợp email

### 14. Người dùng đăng bài viết sau khi đăng nhập

* Đăng bài viết trong profile
* <https://www.tiny.cloud/>
* <https://www.tiny.cloud/docs/tinymce/6/basic-setup/>

### 15. Hiển thị danh sách bài viết của người dùng

* Hiển thị danh sách bài viết của user
* Đẩy duyệt bài viết
* Thống kê bài viết

## Chương 5: Deploy ứng dụng lên server

### 1. Cài đặt môi trường triển khai

* Đơn vị tài trợ VPS: <https://www.mega.vn/>, link đăng ký: <https://my.megacloud.vn/> → Đăng ký tài khoản → Chọn gói VPS → Thanh toán → Chờ khởi tạo VPS → Remote vào VPS
* Cài đặt IIS
* Cài đặt URL Rewrite Module cho IIS: <https://www.iis.net/downloads/microsoft/url-rewrite>
* Cài đặt .NET Runtime trên Server: <https://dotnet.microsoft.com/en-us/download/dotnet/8.0>
* Cài MSSQL: <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>

### 2. Deploy các ứng dụng

* Tạo các website trên IIS, trỏ domain từ DNS vào domain **blog.tedu.work**
* Phân quyền ghi cho upload image
* Cấu hình đa môi trường cho các ứng dụng
* Cấu hình HTTPS qua phần mềm get HTTPS miễn phí.
* Run migration tạo database
* Tạo script database (Script-Migration -from <Migration\_Name>
* Chạy thử ứng dụng

### 3. Kết thúc khóa học

* Sau khi kết thúc, các update sửa lỗi sẽ được thực hiện qua **Pull request** vào nhánh **master** để mọi người nắm được.
* Github: [**https://github.com/teduinternational/tedu-blog**](https://github.com/teduinternational/tedu-blog)
* Link Discord: <https://discord.gg/7D7sTJ4jX9> → channel: **#tedu53-net8-angular**
* **Vấn đề:**
  + Lấy SMTP gửi mail: Dùng SMTP của các bạn
  + Một số lôi UI
  + Một số lỗi của admin

# Tác giả

Tác giả: Toàn Bạch

# Đối tượng học tập

* Sinh viên chuẩn bị tốt nghiệp, đang học MVC, làm project
* Người đi làm muốn học cách xây dựng hệ thống hoàn chỉnh và học hỏi thêm kỹ năng mới

# Yêu cầu tối thiểu

* Sử dụng được C# căn bản, HTML CSS và JavaScript căn bản
* SQL Server căn bản
* ASP.NET Core căn bản

# Thời gian dự kiến ra mắt

* Tháng 11/2023 tại TEDU.COM.VN

# Chính sách học phí

* Học phí: **1.800.000** VNĐ
* Chính sách ra mắt dành cho 50 người đầu tiên: giảm giá 70% còn 540.000 VNĐ