MVC — MVVM (Model-View-ViewModel)

~~Startup.cs~~

Razor pages — for simple pages to receive & display data

Startup.cs

```
Startup
  └→ configuration
  └→ configure (app, env)
        |→ if development
        |→ https redirect
        |→ use static files
        |→ use Routing
        |→ use Authorization
        |→ use Endpoints
```

cshtml

```
@page
<___________>
<___ @Data... >
```

filename.cshtml

```
@page
@using namespace-name
@model model-name
<>
< >
    Model - .
<>
```

filename.cshtml.cs

```
namespace namespace-name {
    public class model-name : PageModel {
    }
```

framework provided Services

→ IHost Appl^n lifetime — logging

→ IHost lifetime

→ IHostEnvironment — appname, envt name, content root path

Host Configuration

Configure HostConfiguration on IHostBuilder

ASPNETCORE - ENVIRONMENT ⇒ environment

ConfigureHostConfiguration (config Host ⇒

→ SetBasePath

→ addJsonfile (Hostsetting)

→ AddEnvt variable

→ add command line

App configu

App Name — key, type, default + entry point, Envt variable

Content root . - key, type, . set value, Default - app assembly

Envt name - key, type default : production, val. envt var

WebApps

capture startup errors

UseStartup <Startup>.()

Program

→ Host. CreateDefault Builder(args).
    .ConfigureWebHostDefaults (webBuilder =>

→ CreateDefaultBuilder

- → Chaind ConfiguratinProvider — source to app config.
- [app-config] - custom proper & behaviour of APP.
- ~~→ appsetting.json~~ → configuration value
- → Content root — gets current director
- → env. variables. & cli arguments
- → loads app config from .appsettings(json. ...
- → logging provider
    - → console
    - → debug
    - → event Source
    - → event log

Configure WebHostDefaults

- → host config
- → kestrel
    - → adds middleware
- → IIS Integration

·cshtml – html markup & C# code with Razor Syntax

·cshtml·cs – C# code for page events

wwwroot — contains assets like HTML, JS, & CSS files

appsettings.json – configdata & connection strings

program.cs – entrypoint for the system

startup.cs – code for configure app behaviour.

→ app settings json

→ Program cs

→ Startup cs

---

appsetting.json

→ loggin,
  ↳ log level
    ↳ default – ~~microsof~~ Information
    ↳ microsof – warning
    ↳ .

– Allowed Hosts

– Connections strings :- {
      → context : —
  }

Program.cs (AspNetCore, .Hosting, Extensions)

~~Stat~~

↳ CreateHostBuilder(args).

→ IHostBuilder()

~~HostBu~~ ~~Host CreateDefaults~~.

Host. CreateDefaultBuilder (args)

.ConfigureWebHostDefaults (webBuilder =>

{

webbuilder . UseStartup<Startup> ();

}) ;

}

Startup.cs

{ AspNetCore, .Builder, Hosting, . Extension, .configuration

Dependancy Injection, Hosting, entity frameworkcore,

→ Configuration

→ ConfigureServices

→ AddDbContext

→ Configure (AppBuilder App, IwebHost env)

Softwar

· Net Framework - Supports websites, serve, destop apps & more on windows (fast buildy)

. Net Core - crossplatform, websites, services & desktop apps
(modular, light weight & flexible)
→ able to use for different computer.

· Net - free, cross platform openwource for web, mobile, desktop games etc

ASP. Net · - framework for web apps with . Net & C#
web apps, APIS, Raltime., Microservices

→ developer platform made up of tools, prog. lang. & libs to built web apps

- fast & scalable
- secure (XSS) - cross site scriptr
(CSRF) - cross site rq. forgery

ASP. Net MVC - decouple UI, (view) data (model) & Appl. logic controller)

ASP. Net WEBAPI - HTTP servies with easy way that returns only data.

Software development framework

→ foundation for developers to build , Software application

→ It includes libraries, packages, compiler etc

→ It improves efficiency of working on the important details of app instead of working on the fundamental parts.

.Net

→ is a software framework that runs primarily on Microsoft windows

→ has a large library that supports interoperability (ability to exchange info)

→ across several languages., platform or build.

→ Brief

.Net → the ecosystem – collection of diff software projects

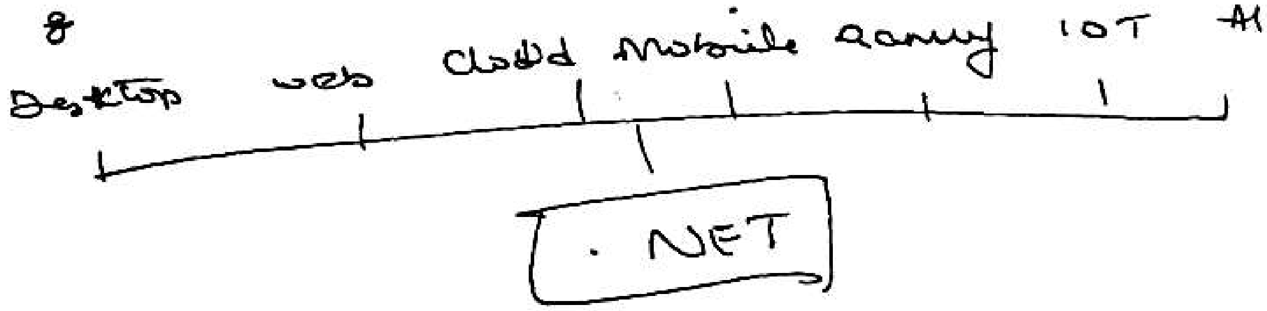
CLR → JVM

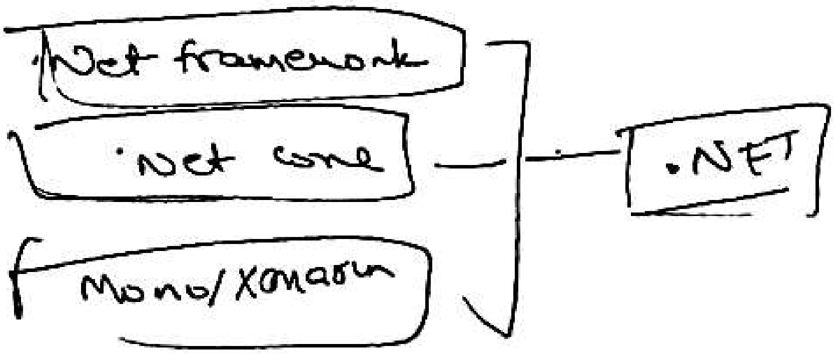
C#, F#, VB → languages

Nuget → npm etc

dotnet cli → entry pt, sdk, javac etc

of





Ecosystem —— Languages - C#, F#, VB
runtime - CLR
Libraries - Base Class Lib, etc

. dll - dynamic link library -

. exe - executable

.Net 5.0. — follow up .Net Core 3.1 & added functionality
— & removed core part

TFM - Target framework moniker

C# — 9.0

---

dotnet - package manager

.Net framework → window applⁿs.
→ WPF, ASP.net form,

.Net Core → crossplatform software framework
→ Modular,

.Net 5.0 - unified framework

URLs

←

useURLs (———)

webroot

.useWeb Root ("public")

(web API)

→ useDeveloperExceptionPage

→ use swagger

→ addSwaggerGen — Author, licence description

→ use HttpsRedirect

→ use Routing

→ Use Authorization

→ use Endpoints

ASP.Net ~~Web API~~ MVC

ASP.Net Web API (dotnet)

→ Startup.cs — Startup & middleware configuration

↳ requests & responses

→ mywebApp/pages — contains web pages for appln

→ mywebapp.csproj — libraries referenced to

~~dotnet~~.

dotnet new webApp -o mywebApp --no-https

~~dotnet~~ dotnet watch run

Razor pages