Працювати з сертифікатами, встановленими в систему, можна за допомогою менеджера сертифікатів (Win + R -> certmgr.msc).

Для запуску додатку був використаний Kestrel сервер, при створенні солюшена був використаний шаблон, який включає в себе базові налаштування для HTTPS. Kestrel автоматично підтримує SNI.

app.UseHsts(); - підключає HSTS, початкове значення 30 днів

app.UseHttpsRedirection(); - підключає redirection HTTP -> HTTPS

HTTPS сертифікат для розробки встановлюється разом з .NET Core SDK, для підтвердження його валідності для локальної розробки потрібно запустити команду

dotnet dev-certs https –trust

Можна згенерувати та використати свій сертифікат за допомогою Powershell утиліти New-SelfSignedCertificate, наприклад наступною командою

New-SelfSignedCertificate -NotBefore (Get-Date) -NotAfter (Get-Date).AddYears(1) -Subject "localhost" -KeyAlgorithm "RSA" -KeyLength 2048 -HashAlgorithm "SHA256" -CertStoreLocation "Cert:\CurrentUser\My" -KeyUsage KeyEncipherment -FriendlyName "HTTPS development certificate" -TextExtension @("2.5.29.19={critical}{text}","2.5.29.37={critical}{text}1.3.6.1.5.5.7.3.1","2.5.29.17={critical}{text}DNS=localhost")

І зробити цей сертифікат довіреним: (Win + R) -> certmgr.msc -> Personal/Certificates, знайти свій сертифікат та скопіювати його в Trusted Root Certification Authorities/Certificates

Код для завантаження сертифікату міститься в класі Program.cs, він дістається з локального стору. Сертифікат також можна зберігати в окремих сервісах (AWS Certificate Manager, Azure Key Management System).

Були вибрані версії TLS 1.3 та 1.2, так як багато браузерів н підтримують версію 1.3. В майбутньому краще використовувати тільки 1.3, так як у версії 1.3 швидший TLS handshake, виключено ненадійні cipher suits з попередніх версій, алгоритми шифрування та аутентифікації об’єднали в AEAD алгоритм. Також всі алгоритми хешування мають бути на основі HMAC.

Для cipher suits були вибрані ті, які рекомендовані в TLS 1.3

* TLS\_AES\_256\_GCM\_SHA384
* TLS\_CHACHA20\_POLY1305\_SHA256,
* TLS\_AES\_128\_GCM\_SHA256,
* TLS\_AES\_128\_CCM\_8\_SHA256,
* TLS\_AES\_128\_CCM\_SHA256

(Kestrel не має адекватних імплементацій обмеження cipher suits для Windows, тому прийшлось алгоритми окремо)

Чому були вибрані ці cipher suits:

AES в режимах GCM і CCM – безпечні варіації блочного алгоритму для симетричного шифрування,

CHACHA20\_POLY1305 – безпечний алгоритм для потокового шифрування

SHA-256, SHA385 – використовується для MAC при хендшейку, при умові що використовується AEAD алгоритм шифрування

