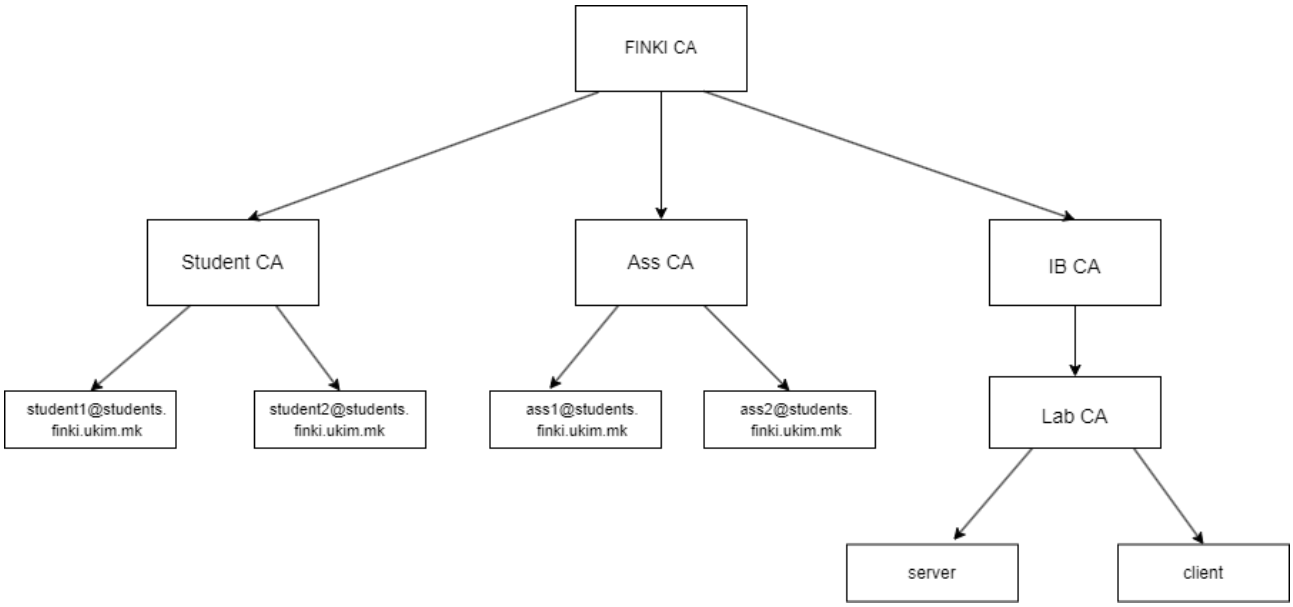
Homework 4 documentation:

GOAL: create this PKI infrastructure and configure my app to work with the server certificate



Guide: https://jamielinux.com/docs/openssl-certificate-authority/index.html

Something worth mentioning here is that im using a **Windows** computer.

Generate a self-signed Root Certificate Authority (FINKI CA)

- In Windows PoweShell (C:\cert_conf\root\FINKI_CA):

```
New-Item -ItemType Directory -Path ".\certs"

New-Item -ItemType Directory -Path ".\crl"

New-Item -ItemType Directory -Path ".\newcerts"

New-Item -ItemType Directory -Path ".\private"

New-Item -ItemType File -Path .\index.txt
```

- In GitBash:

```
#I'm writing in GitBash beacause I encountered some errors with the serial file in PowerShell

echo 00 > serial
```

- Prepared the configuration file. Copied it from https://jamielinux.com/docs/openssl-certificate-authority/appendix/root-configuration-file.html and made changes to the file locations and names and specified some defaults.

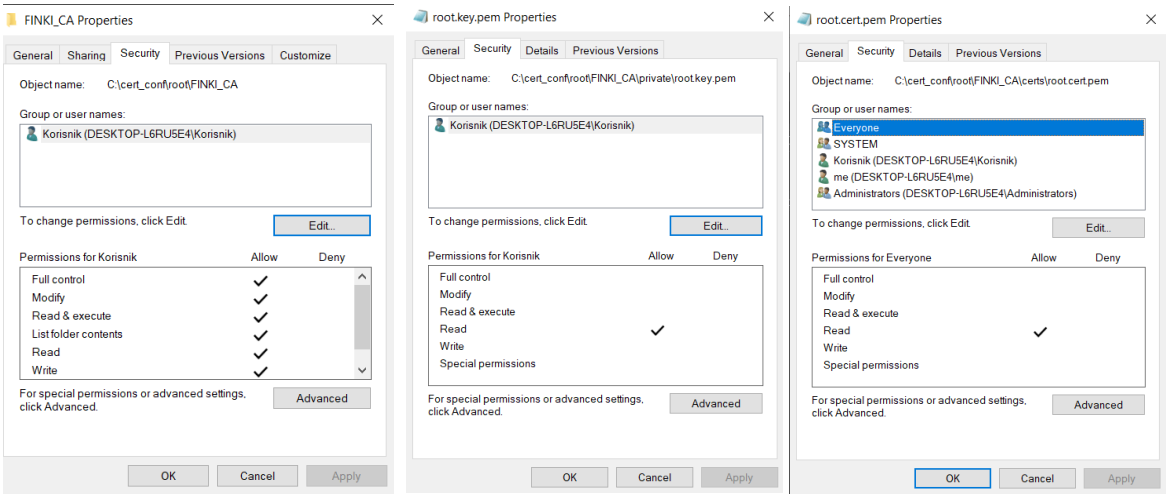
- In Windows PoweShell (C:\cert_conf\root\FINKI_CA):

```
# Create the root key (will use pass phrase: theDOORisLOCKED4!!!)

openssl genrsa -aes256 -out private/root.key.pem 4096

# Create the root certificate

openssl req -config openssl.cnf `

    -key private/root.key.pem `

    -new -x509 -days 7300 -sha256 -extensions v3_ca `

    -out certs/root.cert.pem

# Verify the root certificate

openssl x509 -noout -text -in certs/root.cert.pem
```

- Output:

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            31:37:1e:1a:b2:c0:a6:1d:68:3d:db:13:5d:99:d4:89:d0:61:48:50
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = MK, ST = Skopje, L = Skopje, O = FINKI, OU = IB, CN = IB Project Root CA
        Validity
            Not Before: Jan  7 01:00:59 2024 GMT
            Not After : Jan  2 01:00:59 2044 GMT
        Subject: C = MK, ST = Skopje, L = Skopje, O = FINKI, OU = IB, CN = IB Project Root CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (4096 bit)
```

- Changed permissions:



Generate of first Intermediate Certificate Authority (IB CA), which is signed by FINKI CA, and signs by Lab CA

- In Windows PoweShell (C:\cert_conf\root\FINKI_CA\IB_CA):

```
New-Item -ItemType Directory -Path ".\certs"
```

```
New-Item -ItemType Directory -Path ".\crl"

New-Item -ItemType Directory -Path ".\newcerts"

New-Item -ItemType Directory -Path ".\private"

New-Item -ItemType Directory -Path ".\csr"

New-Item -ItemType File -Path .\index.txt
```

- In GitBash:

```
echo 01 > serial
```

- In Windows PoweShell (C:\cert_conf\root\FINKI_CA\IB_CA):

```
echo 1000 | Out-File -FilePath "C:\cert_conf\root\FINKI_CA\IB_CA\crlnumber" -Encoding ASCII
```

- Prepared the configuration file. Copied it from https://jamielinux.com/docs/openssl-certificate-authority/appendix/intermediate-configuration-file.html and made changes to the file locations and names and specified some defaults.
- In Windows PoweShell (C:\cert_conf\root\FINKI_CA):

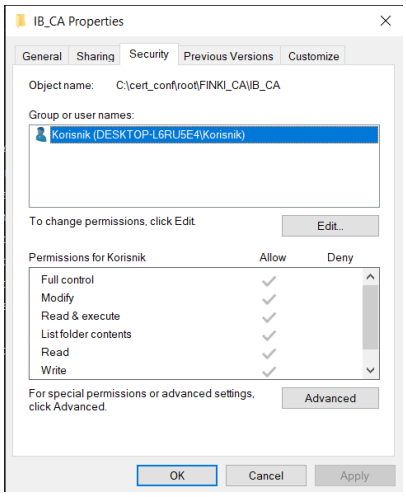
```
# Create intermediate key (will use pass phrase: theDOORisLOCKED4!!!)

openssl genrsa -aes256 -out IB_CA/private/ib.key.pem 4096

# Create the intermediate certificate

openssl req -config IB_CA/openssl.cnf -new -sha256 `

    -key IB_CA/private/ib.key.pem `

    -out IB_CA/csr/ib.csr.pem

openssl ca -config openssl.cnf -extensions v3_intermediate_ca `

    -days 3650 -notext -md sha256 `

    -in IB_CA/csr/ib.csr.pem `

    -out IB_CA/certs/ib.cert.pem

# Verify the root certificate

openssl x509 -noout -text -in IB_CA/certs/ib.cert.pem

openssl verify -CAfile certs/root.cert.pem `

    IB_CA/certs/ib.cert.pem
```

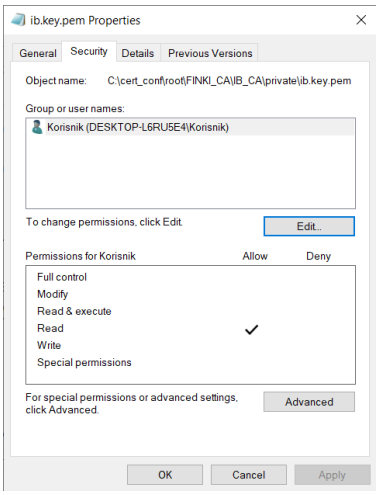
- Output:

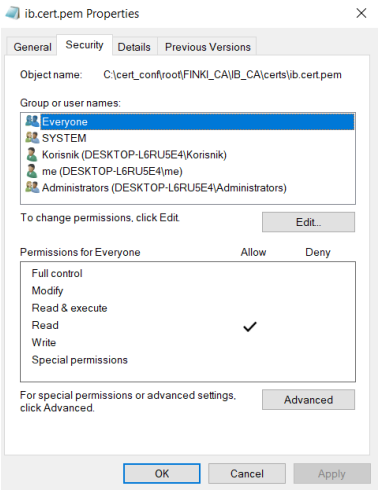
```
Version: 3 (0x2)
Serial Number: 0 (0x0)
Signature Algorithm: sha256WithRSAEncryption
Issuer: C = MK, ST = Skopje, L = Skopje, O = FINKI, OU = IB, CN = IB Project Root CA
Validity
    Not Before: Jan  7 02:26:49 2024 GMT
    Not After : Jan  4 02:26:49 2034 GMT
Subject: C = MK, ST = Skopje, O = FINKI, OU = IB, CN = IB CA
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
        RSA Public-Key: (4096 bit)
```

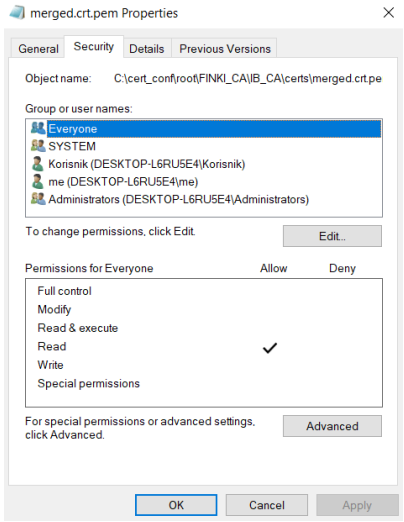
```
>>        IB_CA/certs/ib.cert.pem
IB_CA/certs/ib.cert.pem: OK
```

- Changed permissions:









index.txt - Notepad

File Edit Format View Help

```
V       340104022649Z           00      unknown /C=MK/ST=Skopje/O=FINKI/OU=IB/CN=IB CA
```

Generate a second Intermediate Certificate Authority (Lab CA), which is signed by the IB CA, signs the client and server certificates

- In Windows PoweShell (C:\cert_conf\root\FINKI_CA\IB_CA\Lab_CA):

```
New-Item -ItemType Directory -Path ".\certs"

New-Item -ItemType Directory -Path ".\crl"

New-Item -ItemType Directory -Path ".\newcerts"

New-Item -ItemType Directory -Path ".\private"

New-Item -ItemType Directory -Path ".\csr"

New-Item -ItemType File -Path .\index.txt
```

- In GitBash:

```
echo 01 > serial
```

- In Windows PoweShell (C:\cert_conf\root\FINKI_CA\IB_CA\Lab_CA):

```
echo 1000 | Out-File -FilePath "C:\cert_conf\root\FINKI_CA\IB_CA\Lab_CA\crlnumber" -Encoding ASCII
```

- Prepared the configuration file. Copied it from https://jamielinux.com/docs/openssl-certificate-authority/appendix/intermediate-configuration-file.html and made changes to the file locations and names and specified some defaults.
  Since version 58, Chrome requires SSL certificates to use SAN (Subject Alternative Name) instead of the popular Common Name (CN), thus CN support has been removed (this follows a similar change in Firefox 48). More about this issue here: https://textslashplain.com/2017/03/10/chrome-deprecates-subject-cn-matching/ and here: https://alexanderzeitler.com/articles/Fixing-Chrome-missing_subjectAltName-selfsigned-cert-openssl/. In order to avoid the error NET::ERR_CERT_COMMON_NAME_INVALID, in the configuration file I included alternative names configuration.

- In Windows PoweShell (C:\cert_conf\root\FINKI_CA):

```
# Create intermediate key (will use pass phrase: theDOORisLOCKED4!!!)

openssl genrsa -aes256 -out IB_CA/Lab_CA/private/lab.key.pem 4096

# Create the intermediate certificate

openssl req -config IB_CA/Lab_CA/openssl.cnf -new -sha256 `

    -key IB_CA/Lab_CA/private/lab.key.pem `

    -out IB_CA/Lab_CA/csr/lab.csr.pem
```

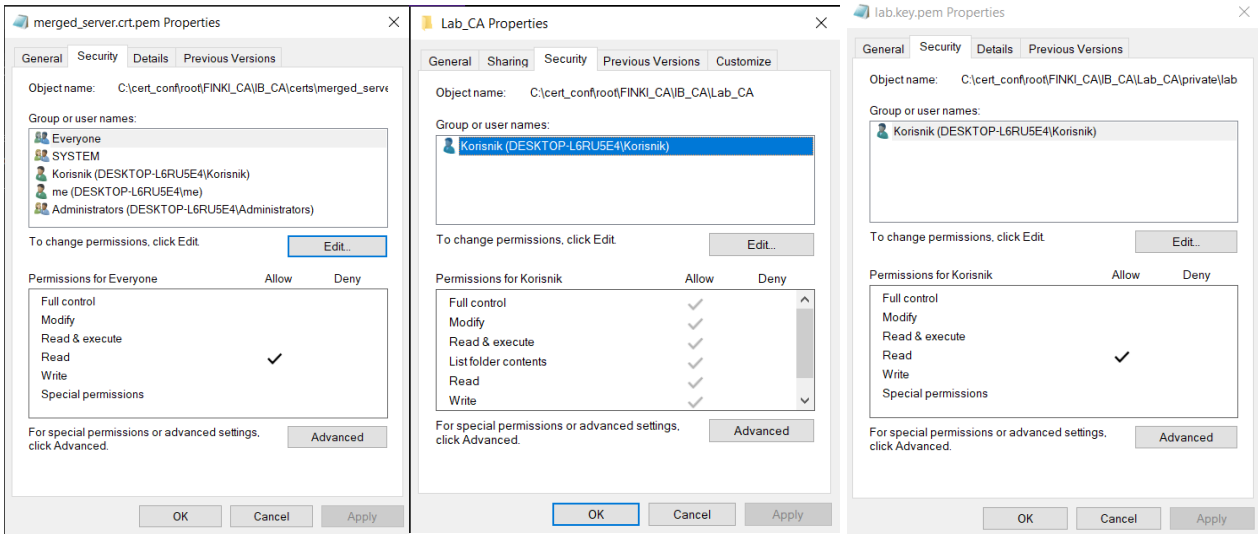
```
openssl ca -config IB_CA/openssl.cnf -extensions v3_intermediate_ca `
    -days 3650 -notext -md sha256 `
    -in IB_CA/Lab_CA/csr/lab.csr.pem `
    -out IB_CA/Lab_CA/certs/lab.cert.pem
# Verify the root certificate
Get-Content .\certs\root.cert.pem, .\IB_CA\certs\ib.cert.pem | Set-Content -Path .\IB_CA\certs\merged.crt.pem
openssl x509 -noout -text -in IB_CA/Lab_CA/certs/lab.cert.pem
openssl verify -CAfile IB_CA/certs/merged.crt.pem `
    IB_CA/Lab_CA/certs/lab.cert.pem
```

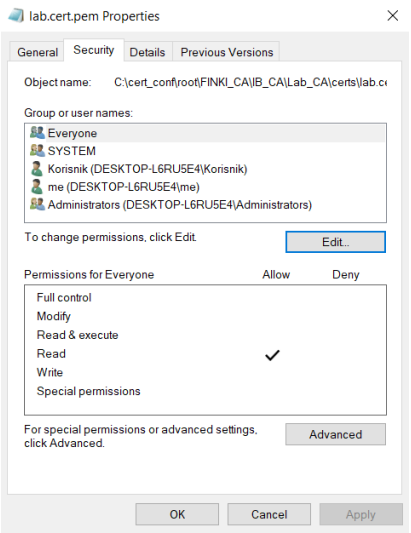
- Output:

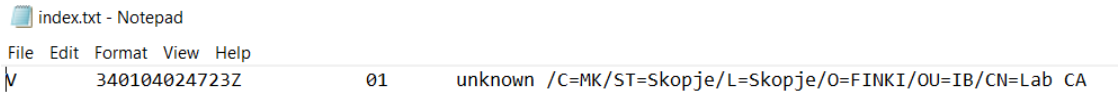
```
Data:
    Version: 3 (0x2)
    Serial Number: 1 (0x1)
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: C = MK, ST = Skopje, O = FINKI, OU = IB, CN = IB CA
    Validity
        Not Before: Jan  7 02:47:23 2024 GMT
        Not After : Jan  4 02:47:23 2034 GMT
    Subject: C = MK, ST = Skopje, L = Skopje, O = FINKI, OU = IB, CN = Lab CA
    Subject Public Key Info:
        Public Key Algorithm: rsaEncryption
            RSA Public-Key: (4096 bit)
```

```
>>        IB_CA/Lab_CA/certs/lab.cert.pem
IB_CA/Lab_CA/certs/lab.cert.pem: OK
```

- Changed permissions:







Generate one server and one client certificate each (with your index)

SERVER:

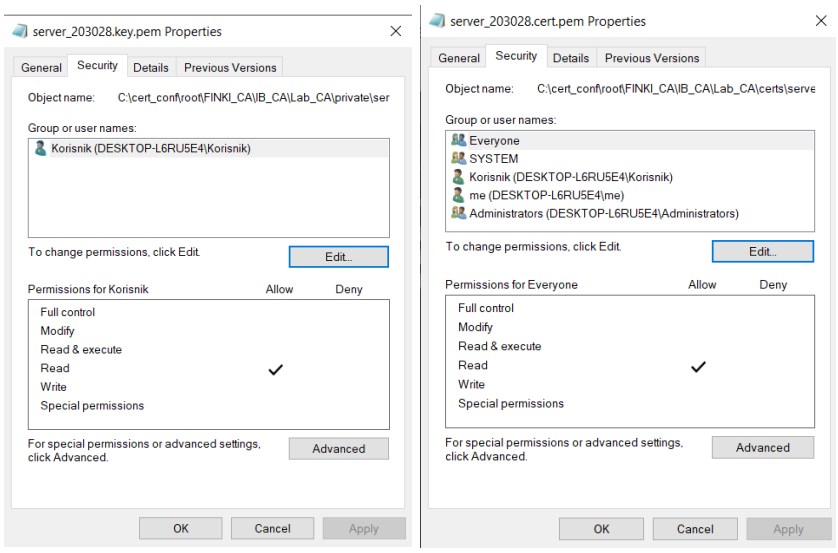
- In Windows PoweShell (C:\cert_conf\root\FINKI_CA):

```
openssl genrsa -aes256 -out .\IB_CA\Lab_CA\private\server_203028.key.pem 2048

openssl req -config .\IB_CA\Lab_CA\openssl.cnf `

    -key .\IB_CA\Lab_CA\private\server_203028.key.pem `

    -new -sha256 -out .\IB_CA\Lab_CA\csr\server_203028.csr.pem

openssl ca -config .\IB_CA\Lab_CA\openssl.cnf `

    -extensions server_cert -days 375 -notext -md sha256 `

    -in .\IB_CA\Lab_CA\csr\server_203028.csr.pem `

    -out .\ib_ca\lab_ca\certs\server_203028.cert.pem

openssl x509 -noout -text `

    -in .\IB_CA\Lab_CA\certs\server_203028.cert.pem

Get-Content .\IB_CA\certs\merged.crt.pem, .\IB_CA\Lab_CA\certs\lab.cert.pem | Set-Content -Path
.\ib_ca\certs\merged_server.crt.pem

openssl verify -CAfile .\IB_CA\certs\merged_server.crt.pem .\IB_CA\Lab_CA\certs\server_203028.cert.pem
```

- Output:

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = MK, ST = Skopje, L = Skopje, O = FINKI, OU = IB, CN = Lab CA
        Validity
            Not Before: Jan  7 02:59:52 2024 GMT
            Not After : Jan 16 02:59:52 2025 GMT
        Subject: C = MK, ST = Skopje, L = Skopje, O = FINKI, OU = IB, CN = localhost
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (2048 bit)
```

```
.\IB_CA\Lab_CA\certs\server_203028.cert.pem: OK
```



CLIENT:

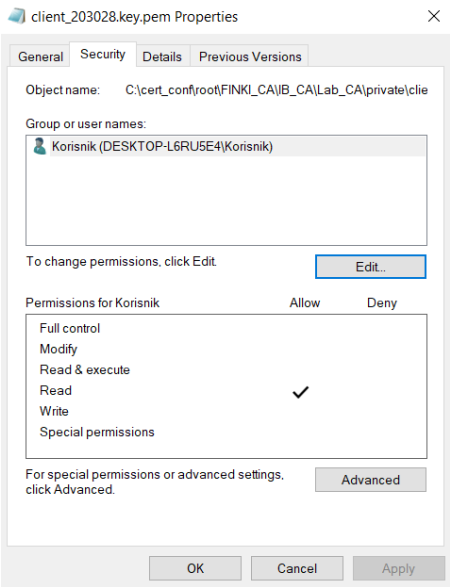
- In Windows PoweShell (C:\cert_conf\root\FINKI_CA):

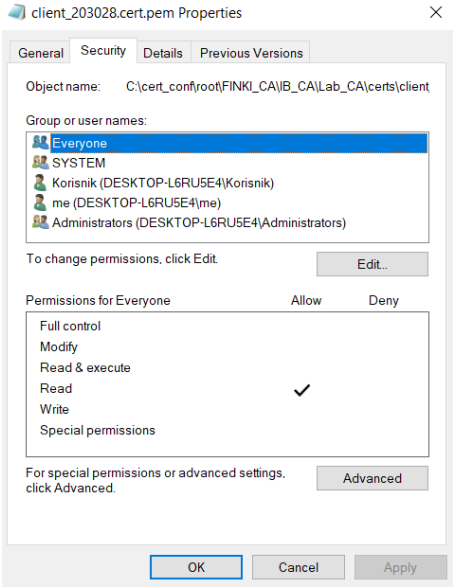
```
openssl genrsa -aes256 -out .\IB_CA\Lab_CA\private\client_203028.key.pem 2048

openssl req -config .\IB_CA\Lab_CA\openssl.cnf `

    -key .\IB_CA\Lab_CA\private\client_203028.key.pem `

    -new -sha256 -out .\IB_CA\Lab_CA\csr\client_203028.csr.pem

openssl ca -config .\IB_CA\Lab_CA\openssl.cnf `

    -extensions usr_cert -days 375 -notext -md sha256 `

    -in .\IB_CA\Lab_CA\csr\client_203028.csr.pem `

    -out .\ib_ca\lab_ca\certs\client_203028.cert.pem

openssl x509 -noout -text `

    -in .\IB_CA\Lab_CA\certs\client_203028.cert.pem

openssl verify -CAfile .\IB_CA\certs\merged_server.crt.pem .\IB_CA\Lab_CA\certs\client_203028.cert.pem
```

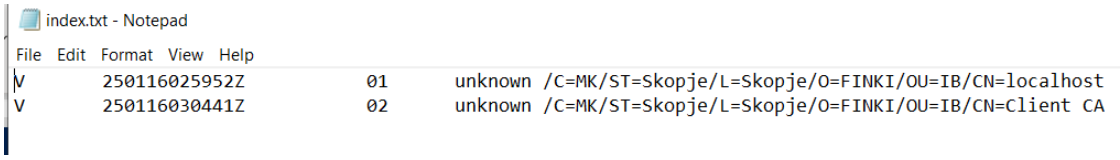
Output:

```
Version: 3 (0x2)
Serial Number: 2 (0x2)
Signature Algorithm: sha256WithRSAEncryption
Issuer: C = MK, ST = Skopje, L = Skopje, O = FINKI, OU = IB, CN = Lab CA
Validity
    Not Before: Jan  7 03:04:41 2024 GMT
    Not After : Jan 16 03:04:41 2025 GMT
Subject: C = MK, ST = Skopje, L = Skopje, O = FINKI, OU = IB, CN = Client CA
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
        RSA Public-Key: (2048 bit)
```

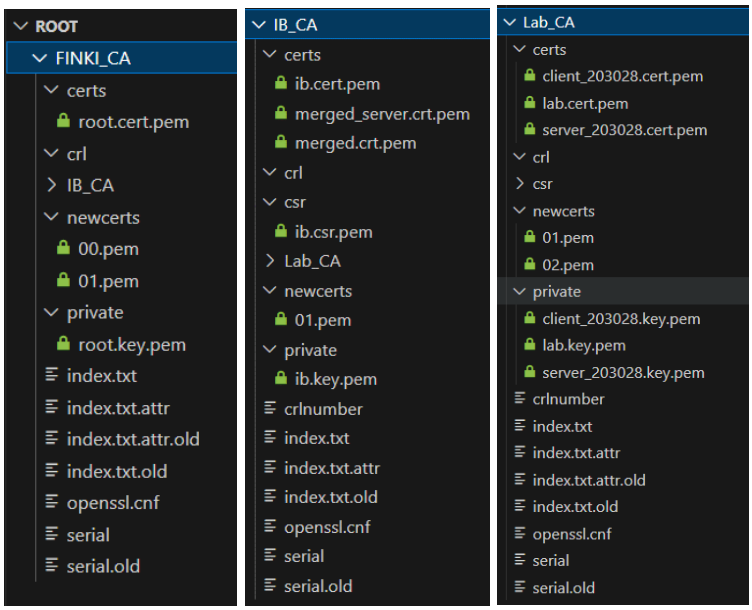
```
.\IB_CA\Lab_CA\certs\client_203028.cert.pem: OK
```







Final hierarchy:



Install a suitable web server (Apache/Nginx, IIS) for your web application from the transition exercises and in doing so:

1. deploy your web application to the server

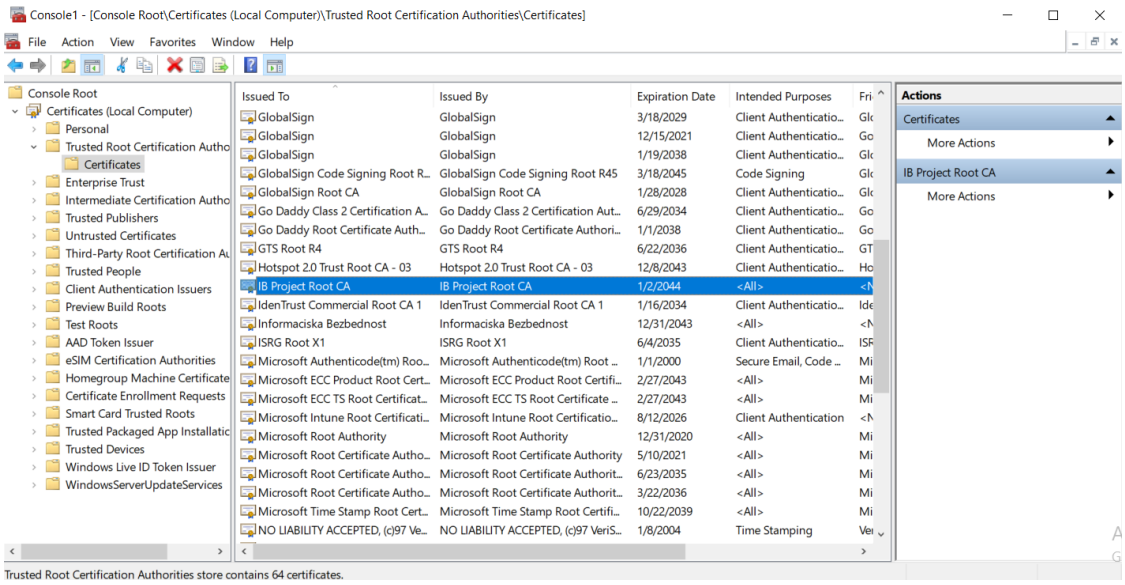
For development purposes, Django comes with its own development server, which is a lightweight server suitable for testing. I can use that built-in development server by running the **python manage.py runserver** command.

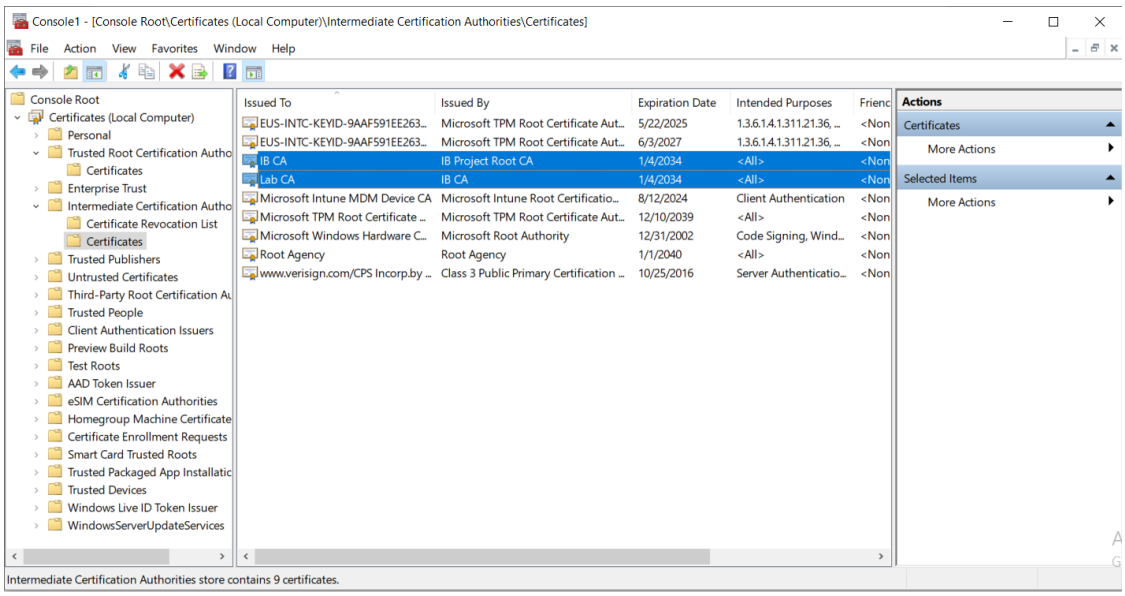
To run my Django development server with SSL support, first I installed **sslserver** by running **pip install django-sslserver** so that I can use the **runsslserver** command provided by the **django-sslserver** package. Then I made small changes in settings.py in my Djagno app:

```
DEBUG = False

ALLOWED_HOSTS = ['localhost', '127.0.0.1']

INSTALLED_APPS = [
    …
    'sslserver',
]

SECURE_SSL_REDIRECT = True
SESSION_COOKIE_SECURE = True
CSRF_COOKIE_SECURE = True
```

2. configure it to work with the server certificate (a green padlock should appear in the address bar :))

I added the root certificate in Trusted Root Certificates in Windows, and I added the intermediate certificates (ib.cert.pem and lab.cert.pem) in Intermediate Certificate Authority in Windows using mmc.exe.





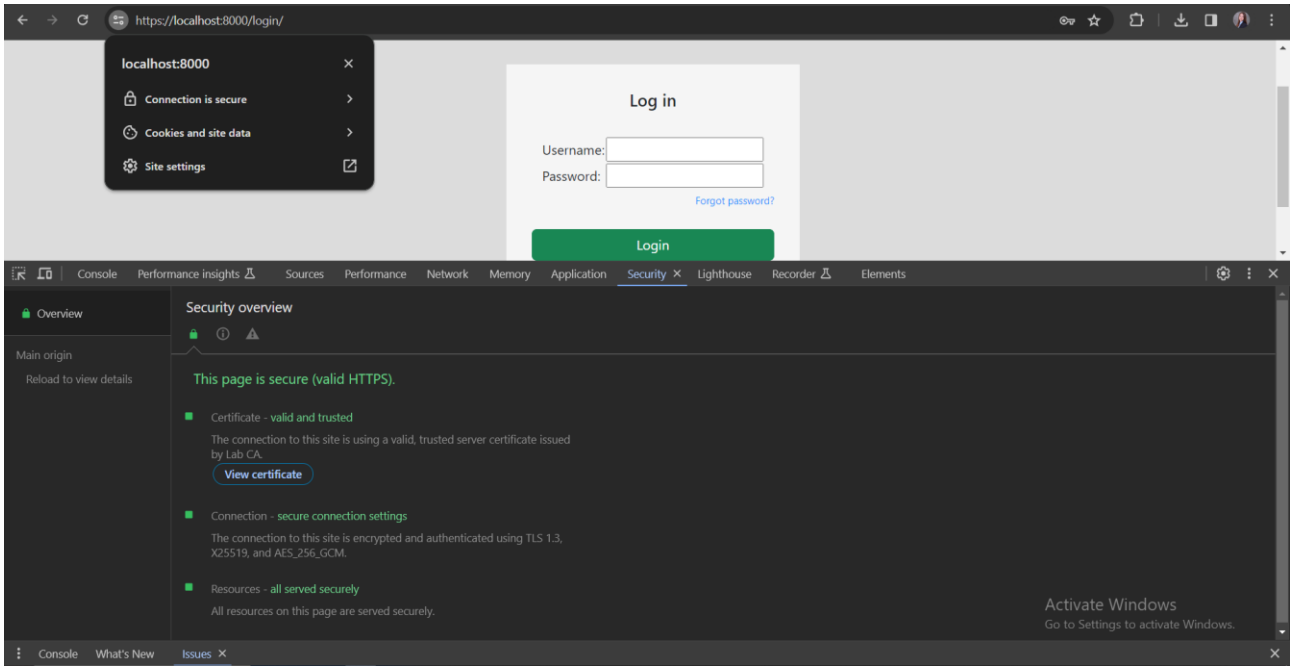
Guide: https://woshub.com/updating-trusted-root-certificates-in-windows-10/

Then I started my application with:

python manage.py runsslserver --cert='C:\\cert_conf\\root\\FINKI_CA\\IB_CA\\Lab_CA\\certs\\server_203028.cert.pem' --key='C:\\cert_conf\\root\\FINKI_CA\\IB_CA\\Lab_CA\\private\\server_203028.key.pem'

The command includes the **--cert** and **--key** options to specify the paths to the server certificate and private key files.

Then I went to https://localhost:8000/login/ where a green padlock appeared (it says that the connection is secure and that the certificate is valid).



Certificate Viewer: localhost

General Details

Issued To

Common Name (CN)	localhost
Organization (O)	FINKI
Organizational Unit (OU)	IB

Issued By

Common Name (CN)	Lab CA
Organization (O)	FINKI
Organizational Unit (OU)	IB

Validity Period

Issued On	Sunday, January 7, 2024 at 3:59:52 AM
Expires On	Thursday, January 16, 2025 at 3:59:52 AM

SHA-256 Fingerprints

Certificate	35679f9701f60851f24c4bb47694f89fe818fec0f3b72ebf7c8ca39936128c51
Public Key	383df0f2f718756ed2f66c945b96711015d576fc9e33177f5c56a1c61b021d2d