Group Members:	Section:	Date:	

ITS115 – Cryptography
Final Project

The **Department of Information and Communications Technology (DICT)**, through the **Philippine National Public Key Infrastructure (PNPKI)**, issues **public key certificates** free of charge.

A public key certificate can be used to **secure a wide range of applications**—such as web connections, email communications, code signing, document verification, user authentication, Internet of Things devices, and financial transactions.

By using public key certificates, you ensure the **confidentiality, integrity, authenticity and non-repudiation** of your information.

For details on DICT's public key certificates and how to apply, visit https://dict.gov.ph/pnpki.

In this project, you will act as a **Certificate Authority (CA)** responsible for issuing public key certificates.

You will use **XCA**—a tool intended for creating X.509 certificates. Your ultimate goal is to **generate an end-entity certificate**—chained back to your CA certificates—that you'll install on your web server to establish a **secure (HTTPS) connection between browser and server**.

Please follow the step-by-step instructions precisely. As you work through the steps, you'll encounter **short-answer questions**. Each question includes a hyperlink to an external resource. Click the link, locate the exact answer in the material, and **copy them exactly as written**. **Different answers will not be accepted.**

I. Group Assignments

Each group will act as a Certification Authority (CA) and use the following organization names:

- Group 1 – **AlphaCert**
- Group 2 – **BetaTrust**
- Group 3 – **GammaShield**
- Group 4 – **DeltaSign**
- Group 5 – **EpsilonMark**
- Group 6 – **ZetaSecure**

II. Requirements

Ensure the latest versions of the following are installed:

- **XCA**
- **Google Chrome**

Short Answer Questions

1. What is a **public key infrastructure (PKI)**?
   Hint: *A set of policies,… and revoke public key certificates.*

2. What is a **public key certificate**?
   Hint: *A digital document…to a public key.*

3. **SSL/TLS** establishes a secure channel between a client (e.g., a web browser) and a server. During the handshake, after the server presents its public key certificate, **what does the client encrypt with the server's public key and send back**?
   Hint: _____ _____

4. What **key** does the server use to decrypt the information sent by the client **in order to generate the session key**?

5. Who issues **digital certificates**?

6. What is **X.509**?

7. What **three components** are included in an X.509 certificate?
   Hint: *Each X.509 certificate includes …issuing certificate authority (CA).*

III. Setting Up Your Certification Authority with XCA

1. **Create Your Project Folder**
   a. Create a folder named **its115_final_project** on your **Desktop**.
2. **Set Up a New Database in XCA**
   a. Open XCA.
   b. Go to **File > Close Database** to close any open databases.
   c. Go to **File > New Database** to create a new one.
   d. Name it: **[Organization Name]CA** (for example, AlphaCertCA).
   e. Save it inside the **its115_final_project** folder.
   f. When prompted, enter a password to encrypt your private keys. To keep it simple, **use your group leader's institutional email as your password** (for example, rey.pollux@marsu.edu.ph).
3. **Change Database Encryption Settings**
   a. When a warning appears ("The currently used PFX/PKCS#12 algorithm 'PBE-SHA1-3DES' is insecure"), click **Change**.

b. In the **XCA Options** window:
    i. Under the Settings tab, set **PKCS#12 encryption algorithm** to **AES-256-CBC**.
    ii. Click **OK**.

4. **Set as Default Database**
    a. Go to **File > Set as Default Database**.

Short Answer Questions

1. In the previous step, a warning message appeared indicating that PBE-SHA1-3DES is insecure as a PKCS#12 encryption algorithm due to **CVE-2016-2183**. As a result, 3DES has been deprecated, and it is recommended to use AES instead.
   What vulnerability is described by CVE-2016-2183?
   Hint: *A flaw was found …used a DES/3DES based ciphersuite.*

2. What is PKCS#12?
   Hint: *It is commonly used …of a chain trust.*

IV. Create a Template for Your CA Certificate

1. **Create a New Template**
    a. Go to **Templates** and click **New Template**.
    b. Select **[default] CA** as the preset template.
2. **Fill in the Subject Information**
    a. In the **Subject** tab of the **Edit XCA template** window, enter the following:
        - **Internal Name: [Organization Name]CA** (for example, AlphaCertCA)
        - **Country Name:** PH
        - **State or Province Name:** Marinduque
        - **Locality Name:** Boac
        - **Organization Name:** Marinduque State University
        - **Organizational Unit Name:** College of Information and Computing Sciences
        - **Email Address:** Group leader's institutional email
2. **Add Additional Distinguished Name**
    a. In the **Subject** tab, click **Add** to add new distinguished name entry.
    b. Change the **Type** from **Country Name** to **Description**.
    c. In the **Content** field, enter the last names of all your group members, starting with the group leader.
3. **Set Certificate Validity**
    a. In the **Time range** section of the **Extensions** tab:
        i. Set validity to **20 years**.
        ii. Check the **Midnight** box so the certificate expires at midnight.
        iii. Click **Apply**.
4. **Set Key Usage**
    a. In the **X509v3 Key Usage (left-side)** section of the **Key usage** tab:

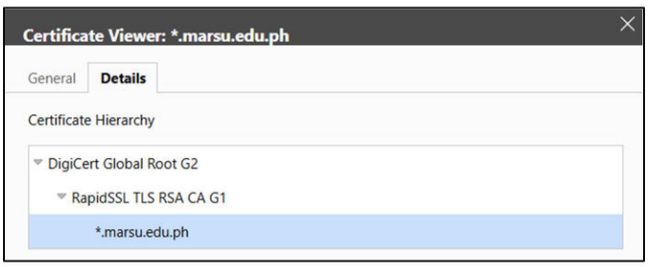
i. Check the **Critical** box.
ii. Select only **Certificate Sign** and **CRL Sign**.

5. **Set Netscape Comment**
   a. In the **Netscape** tab:
      i. Set the Netscape Comment to: **[Organization Name] CA** (for example, AlphaCert CA).
6. **Finish**
   a. Leave other settings as default.
   b. Click **OK**.
   c. You should see a message saying "Successfully created the XCA template".

Information

For both administrative and security-related reasons, X.509 certificates are typically combined into **chains** for validation.

For example, the **SSL/TLS (end-entity) certificate** for **marsu.edu.ph** is signed using one of DigiCert's **intermediate CA certificates**, **RapidSSL TLS RSA CA G1**. In turn, the intermediate CA certificate is signed using **DigiCert Global Root G2**, the **root CA certificate**.

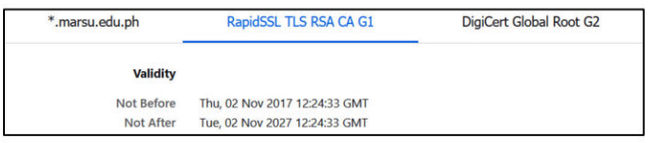


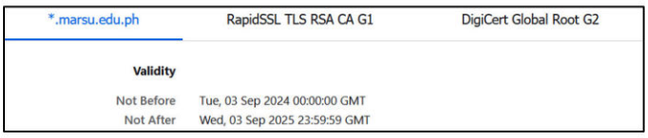
For publicly trusted websites, the web server will provide its own end-entity certificate, plus any intermediates required for validation. The root CA certificate with its public key will be included in the end user's operating system and/or browser application, resulting in a complete **chain of trust**.

Information

Root CA certificates have the longest validity periods, followed by intermediate CA certificates, and then end-entity certificates. **A certificate's validity must fall within the validity period of the certificates above it in the hierarchy.**

In the previous example, the **DigiCert Global Root G2 certificate** is valid for **25 years**, its **intermediate CA certificate** for **10 years**, and its **end-entity certificate** for **1 year**.





Short Answer Questions

1. What certificate is at the **top level of trust** in a certificate authority?

2. What are the **two characteristics** of this certificate?
   Hint: *Their certificates …and web browsers.*

3. What **certificate**, signed by the root CA, issues end-entity certificates?

4. What's the **purpose** of having intermediate CAs?
   Hint: *By segregating duties, …or CRLs.*

V. Create a Root CA Certificate

1. **Start Creating a New Certificate**
   a. Go to **Certificates** and click **New Certificate**.
   b. In the **Source** tab of the **Create x509 Certificate** window:
      i. Select your template in the **Template for the new certificate** section (for example, AlphaCertCA).
   c. Click **Apply all**.
2. **Set the Certificate to Self-Signed**
   a. In the **Signing** section:
      i. Click **Create a self-signed certificate**.
3. **Fill in Subject Details**
   a. In the **Subject** tab, enter the following:
      i. **Internal Name:** [Organization Name]CA - Root (for example, AlphaCertCA - Root)
      ii. **Common Name:** Same as Internal Name
4. **Generate the Private Key**
   a. In the **Private key** section:
      i. Click **Generate a new key**.
      ii. Set Key Type to **RSA** and Key Size to **2048 bits**.
      iii. Click **Create**.
5. **Finish**
   a. Leave other settings as default.
   b. Click **OK**.
   c. **Your Root CA Certificate should now be created.**

VI. Create an Intermediate CA Certificate

1. **Start Creating a New Certificate**
    a. Go to **Certificates** and click **New Certificate**.
    b. In the **Source** tab of the **Create x509 Certificate** window:
        i. Select your template in the **Template for the new certificate** section (for example, AlphaCertCA).
    c. Click **Apply all**.
2. **Sign the Intermediate CA Certificate with the Root CA Certificate**
    a. In the Signing section:
        i. Click **Use this Certificate for signing**.
        ii. Select your **root CA certificate** (for example, AlphaCertCA - Root).
3. **Fill in Subject Details**
    a. In the **Subject** tab, enter the following:
        i. **Internal Name:** [Organization Name]CA - Intermediate (for example, AlphaCertCA - Intermediate)
        ii. **Common Name:** Same as Internal Name
4. **Generate the Private Key**
    a. In the **Private key** section:
        i. Click **Generate a new key**.
        ii. Set Key Type to **RSA** and Key Size to **2048 bits**.
        iii. Click **Create**.
5. **Set Certificate Validity**
   For the intermediate CA certificate, set the validity period shorter than the root CA certificate.
    a. In the **Time range** section of the **Extensions** tab:
        i. Set validity to **10 years**.
        ii. Check the **Midnight** box so the certificate expires at midnight.
        iii. Click **Apply**.
6. **Finish**
    a. Leave other settings as default.
    b. Click **OK**.
    c. **Your Intermediate CA Certificate should now be created.**

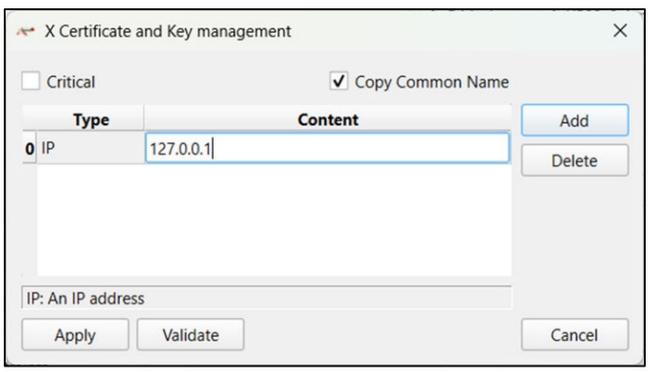
VII. Create a Server Certificate

1. **Start Creating a New Certificate**
    a. Go to **Certificates** and click **New Certificate**.
    b. In the **Source** tab of the **Create x509 Certificate** window:
        i. Select your template in the **Template for the new certificate** section (for example, AlphaCertCA).
    c. Click **Apply all**.
2. **Sign the Server Certificate with the Intermediate CA Certificate**
    a. In the Signing section:
        i. Click **Use this Certificate for signing**.
        ii. Select your **intermediate CA certificate** (for example, AlphaCertCA - Intermediate).
3. **Fill in Subject Details**
    a. In the **Subject** tab, enter the following:
        i. **Internal Name**

Use the **name of the organization you're working with in your capstone project** as the internal name, **formatted like a URL**.
For example, if your capstone project is with **Big Brew Marinduque**, you could set the internal name to **www.bigbrewmarinduque.com.ph**.

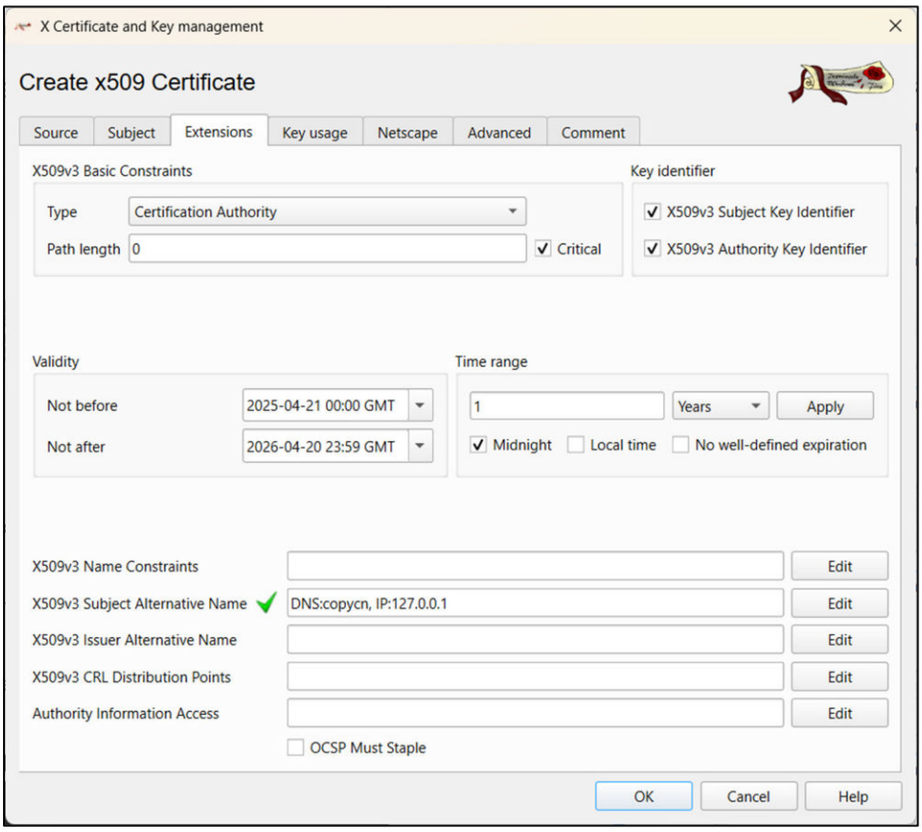
ii. **Common Name:** Same as Internal Name

4. **Generate the Private Key**
   a. In the **Private key** section:
      i. Click **Generate a new key**.
      ii. Set Key Type to **RSA** and Key Size to **2048 bits**.
      iii. Click **Create**.
5. **Set Certificate Validity**
   For the server certificate, set the validity period shorter than the intermediate CA certificate.
   a. In the **Time range** section of the **Extensions** tab:
      i. Set validity to **1 year**.
      ii. Check the **Midnight** box so the certificate expires at midnight.
      iii. Click **Apply**.
6. **Configure Subject Alternative Name**
   a. In the **X509v3 Subject Alternative Name** section of the **Extensions** tab:
      i. Click **Edit**. An **SAN editor** window will appear.
      ii. Check the **Copy Common Name** box.
      iii. Click **Add** to create a new entry.
      iv. Change the **Type** from **URI** to **IP**.
      v. In the **Content** field, enter **127.0.0.1**.



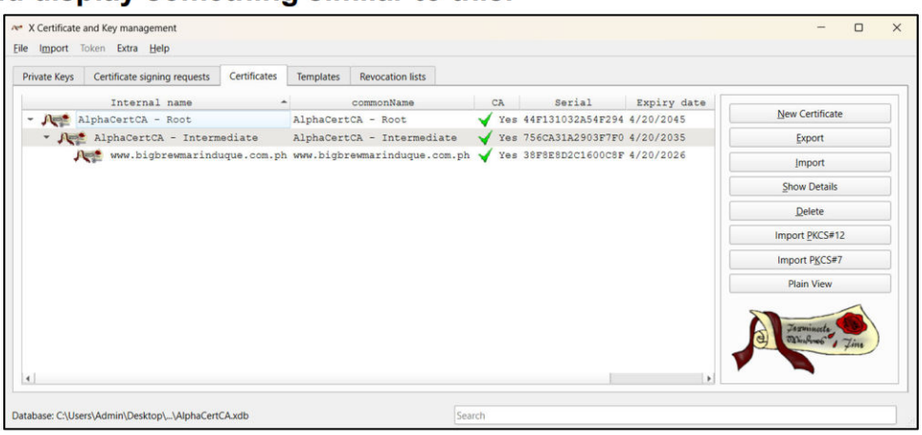
      vi. Click **Validate** to ensure there are no errors.

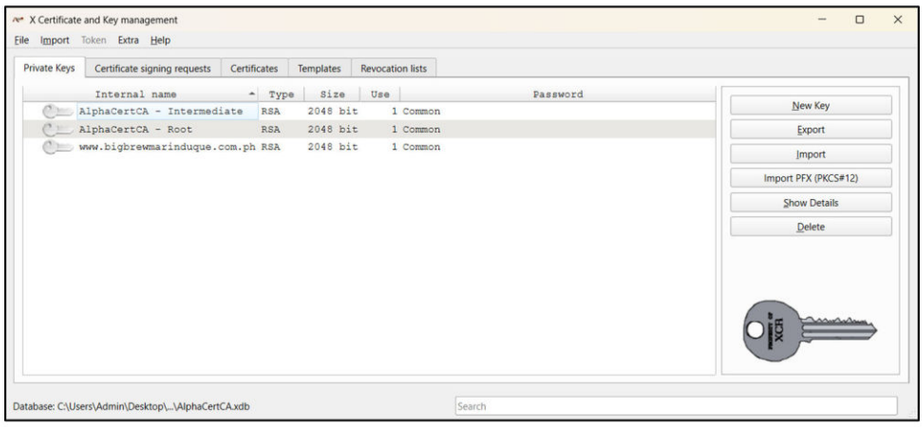
      vii. Click **Apply** to save these changes.
      viii. Back in the **Extensions** tab, you should now see a ✓ next to **X509v3 Subject Alternative Name**, indicating that 127.0.0.1 has been added.



7. **Set Key Usage**
    a. In the **X509v3 Key Usage (left-side)** section of the **Key usage** tab:
        i. Check the **Critical** box.
        ii. Select only **Digital Signature** and **Key Encipherment**.
    b. In the **X509v3 Extended Key Usage (right-side)** section of the **Key usage** tab:
        i. Check the **Critical** box.
        ii. Select only **TLS Web Server Authentication** and **TLS Web Client Authentication**.
8. **Finish**
    a. Leave other settings as default.
    b. Click **OK**.
    c. **Your server certificate should now be created.**

Your XCA should display something similar to this.

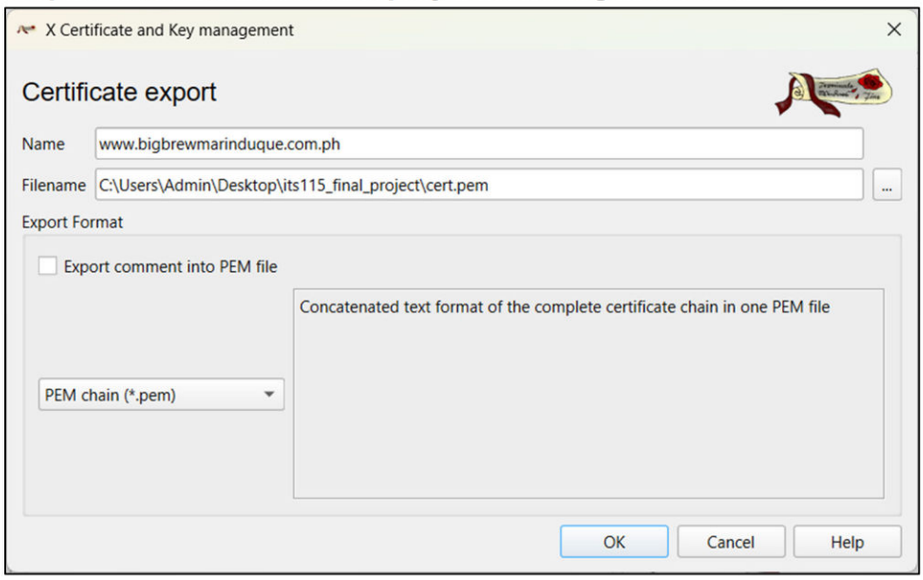




VIII. Export the Server Certificate for Your Website

1. In XCA, go to the **Certificates** tab.
2. Select your **server certificate** (for example, www.bigbrewmarinduque.com.ph).
3. Click **Export** on the right-hand side.
4. In the **Certificate Export** window, choose to save the certificate as a **PEM chain file (*.pem)**.
5. Save the file in the **its115_final_project** folder.
6. Rename the exported file to **cert.pem** to indicate that it is a certificate.

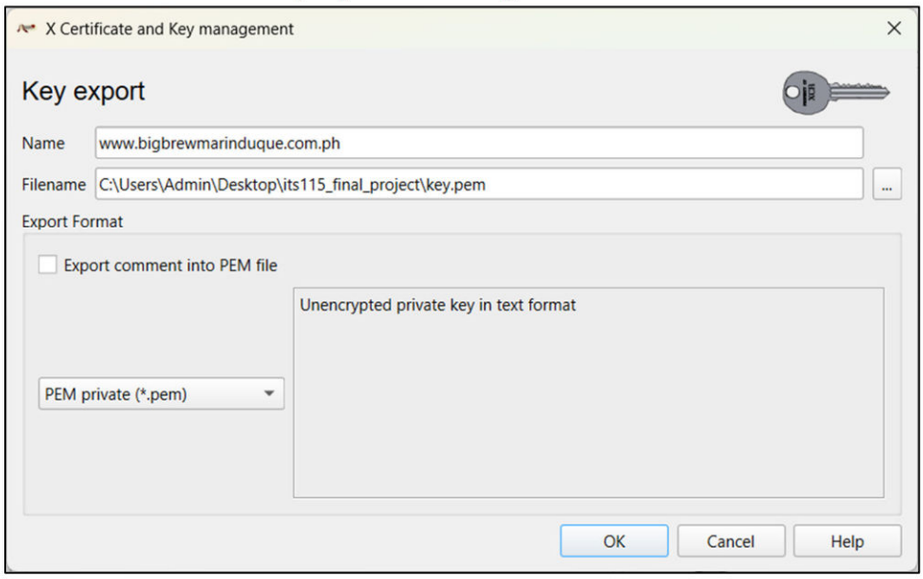
Your Certificate Export window should display something similar to this.



IX. Export the Private Key of the Server Certificate for Your Website

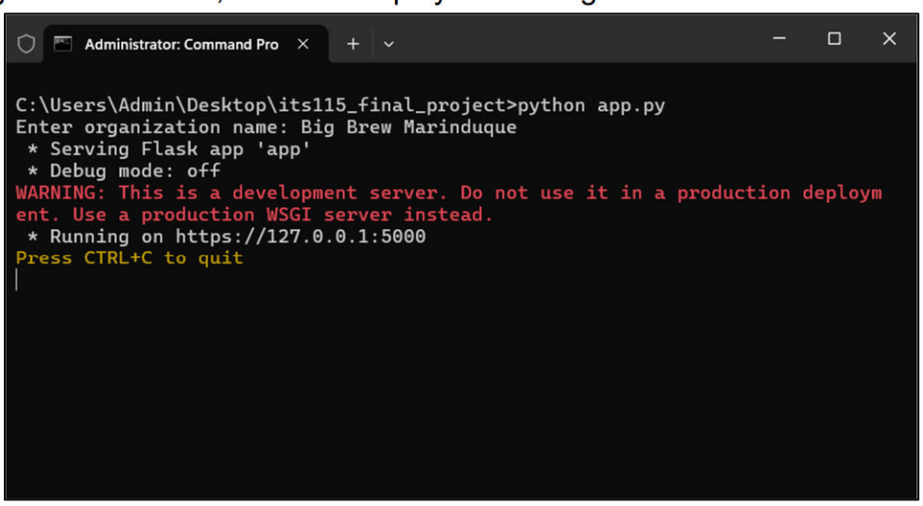
1. In XCA, go to the **Private Keys** tab.
2. Select the **private key associated with your server certificate** (for example., www.bigbrewmarinduque.com.ph).
3. Click **Export** on the right-hand side.
4. In the **Key Export** window, choose to save the private key as a **PEM private file (*.pem)**.
5. Save the file in the **its115_final_project** folder.
6. Rename the exported file to **key.pem** to indicate that it is a private key.

Your Key Export window should display something similar to this.

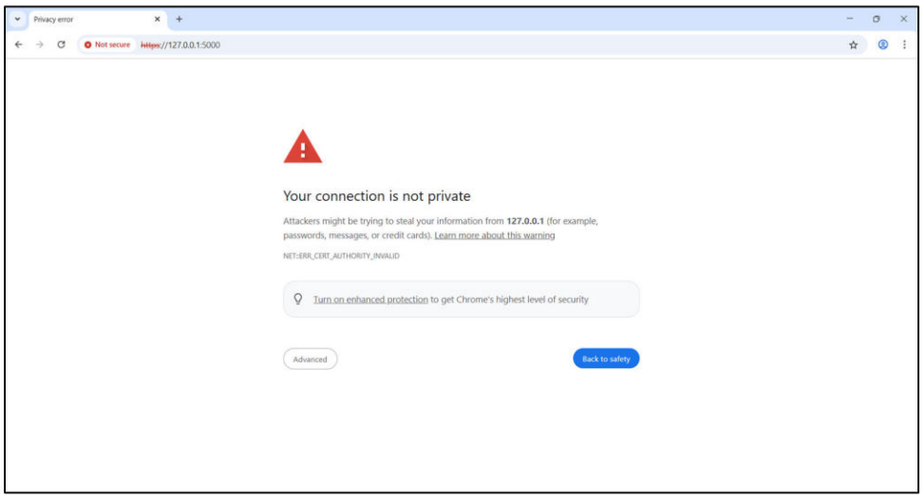


X. Run a Website Using the Server Certificate

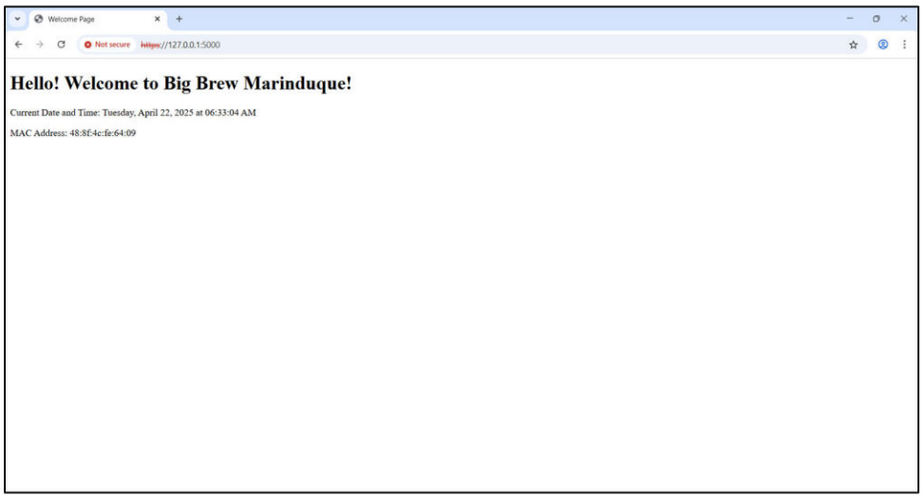
1. Download the executable file named **app.exe** from the provided Google Drive link.
2. Move the app.exe file into the **its115_final_project** folder.
3. Run the app.exe file by double-clicking it.
4. When prompted, enter your chosen organization name (for example, Big Brew Marinduque).
5. After entering the information, it should display something similar to this.



6. In the screenshot above, the website is running at **https://127.0.0.1:5000**.
   Open Google Chrome and enter this address in the browser's address bar to access the website.
7. The browser will warn you that **"Your connection isn't private"** and indicate an **NET::ERR_CERT_AUTHORITY_INVALID** error.



8. Click **Advanced** and **Continue to 127.0.0.1 (unsafe)**.
   The browser should display something similar to this.



9. Take a **screenshot** of the insecure website and save it in the **its115_final_project** folder. Name the screenshot **insecure_website**.

Short Answer Questions

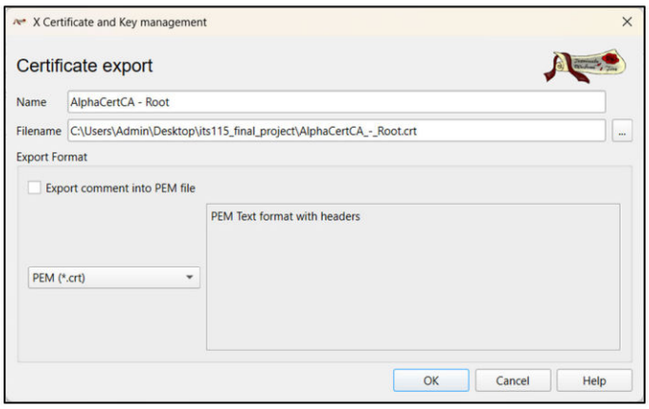
1. What does the **NET::ERR_CERT_AUTHORITY_INVALID** error mean?
   Hint: *Your browser …website's SSL certificate.*

XI. Add the Server Certificate as Trusted by Chrome

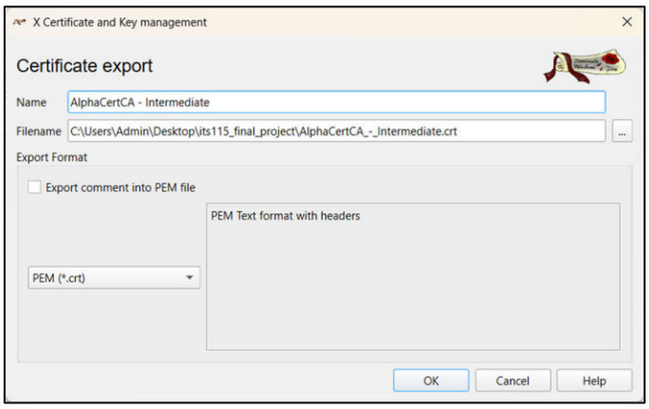
To make Chrome trust your server certificate for localhost (127.0.0.1), follow these steps:

Export the Root and Intermediate Certificates

1. In XCA, go to the **Certificates** tab.
2. Select your **root certificate** (for example, AlphaCertCA - Root).
3. Click **Export** on the right-hand side.
4. In the **Certificate Export** window, choose to save the certificate as a **PEM (*.crt)**.



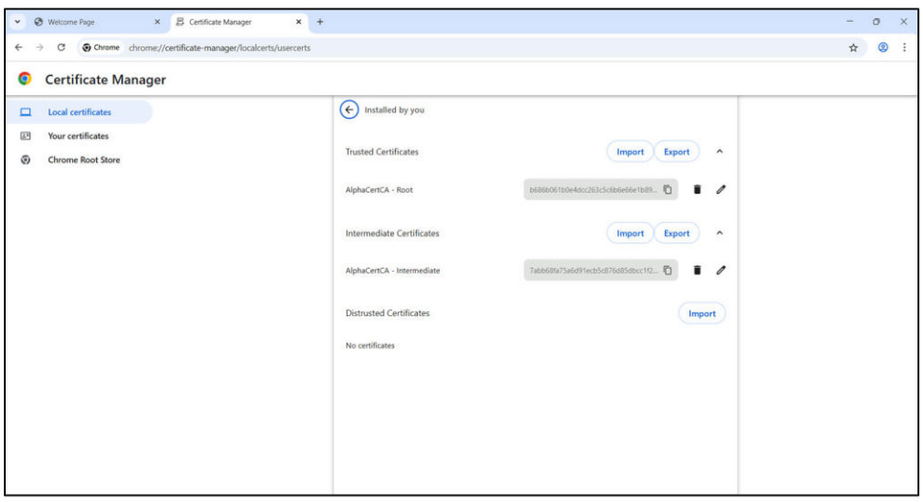
5. Save the file in the **its115_final_project** folder.
6. Next, select your **intermediate certificate** (for example, AlphaCertCA - Intermediate)
7. Click **Export** on the right-hand side.
8. In the **Certificate Export** window, choose to save the certificate as a **PEM (*.crt)**.



9. Save the file in the **its115_final_project** folder.

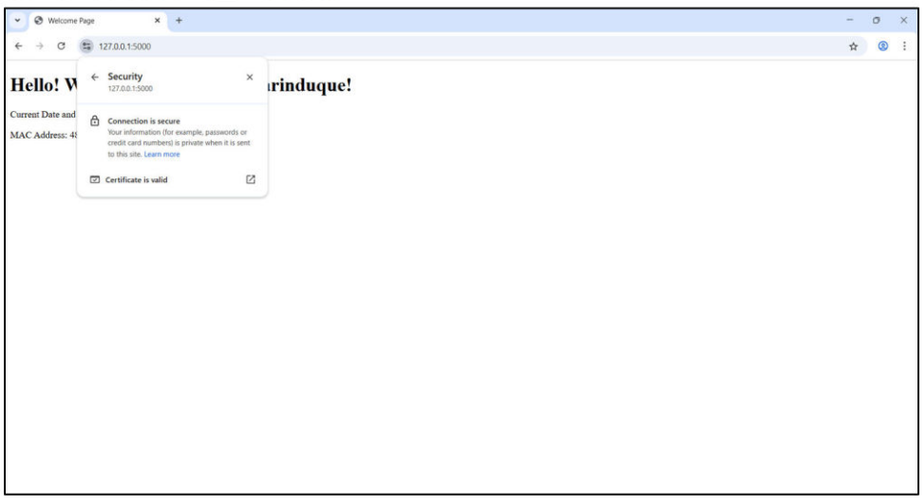
Import the Certificates into Chrome

1. In Chrome, click the menu ⋮ in the top right corner.
2. Go to **Settings > Privacy and security > Security**.
3. Scroll down and click **Manage certificates**.
4. Under **Local Certificates**, go to the **Custom** section and click **Installed by you**.
5. Import your **root certificate** (for example, AlphaCertCA_-_Root.crt) into **Trusted Certificates**.
6. Import your **intermediate certificate** (for example, AlphaCertCA_-_Intermediate.crt) into **Intermediate Certificates**.



Verify the Certificate

1. Clear your browsing data.
2. Close the tab where your website is open.
3. Open a new tab and type in your website's address again.
4. Check the address bar — the **Not Secure** warning should be gone.
5. Click the **View site information** icon to the left of the address bar. Then, click **Connection is secure**.
6. A message should appear showing that your certificate is valid indicating that the connection is secure between the browser and the website server.
7. Take a **screenshot** of your secure website and save it in the **its115_final_project** folder. Name the screenshot **secure_website**.



If you have reached this point, congratulations!

You successfully created a certificate chain — including root, intermediate, and server certificates — and used it to secure a web connection. This shows your understanding of PKI principles and the practical application of SSL/TLS encryption.

XII. Submit Files

Ensure your **its115_final_project** folder contains the following items:

File Description	File Format	Example
1. XCA database	.xdb	AlphaCertCA.xdb
2. Server certificate	.pem	cert.pem
3. Private key of the server certificate	.pem	key.pem
4. Application to run website	.exe	app.exe
5. Screenshot of the insecure website	.png	insecure_website.png
6. Root certificate	.crt	AlphaCertCA_-_Root.crt
7. Intermediate certificate	.crt	AlphaCertCA_-_Intermediate.crt
8. Screenshot of the secure website	.png	secure_website.png

Before submitting, double-check that all the files listed above are present. Once confirmed, compress the entire its115_final_project folder into a **ZIP file**. Only the **group leader** should upload the ZIP file to **Google Classroom**.

XIII. Submit Manual

Save and submit this manual along with your answers. Only the group leader should also upload it to Google Classroom.