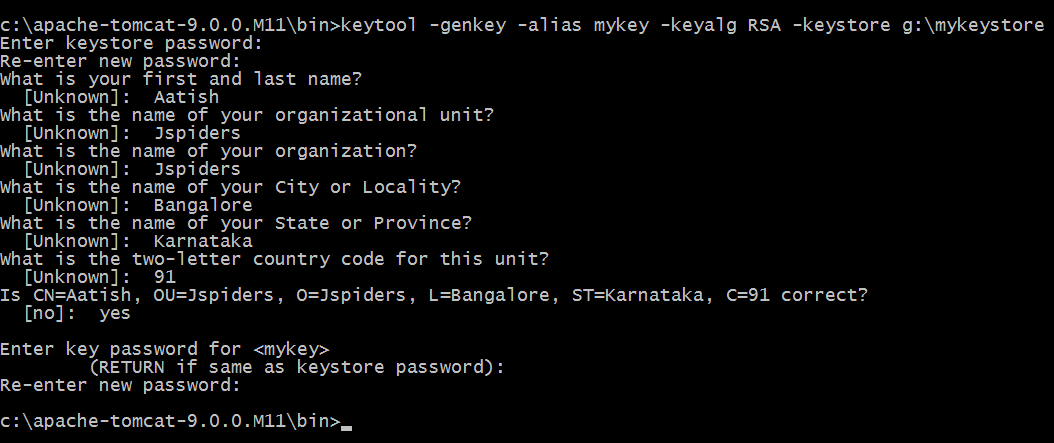
**Enabling SSL or Https Support in Tomcat**

1. **Generate Keystore**

* The first step is to create a self-signed certificate. During the keystore creation process, you need to assign a password and fill in the certificate’s detail.
* Using “keytool” to create a self-signed certificate.

cmd: **keytool -genkey -alias mykey -keyalg RSA -keystore g:\mykeystore**



* Here, you just created a certificate named “**mykeystore**“, which locate at “**g:\**“.

**To Check Certificate Details:**

cmd: **keytool -list -keystore g:\mykeystore**

****

1. **Add a Connector to server.xml:**

<!-- Define a SSL HTTP/1.1 Connector on port 8443

This connector uses the JSSE configuration, when using APR, the

connector should be using the OpenSSL style configuration

described in the APR documentation -->

<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"

maxThreads="150" scheme="https" secure="true"

clientAuth="false" sslProtocol="TLS"

keystoreFile="G:\mykeystore"

keystorePass="Admin@11" />

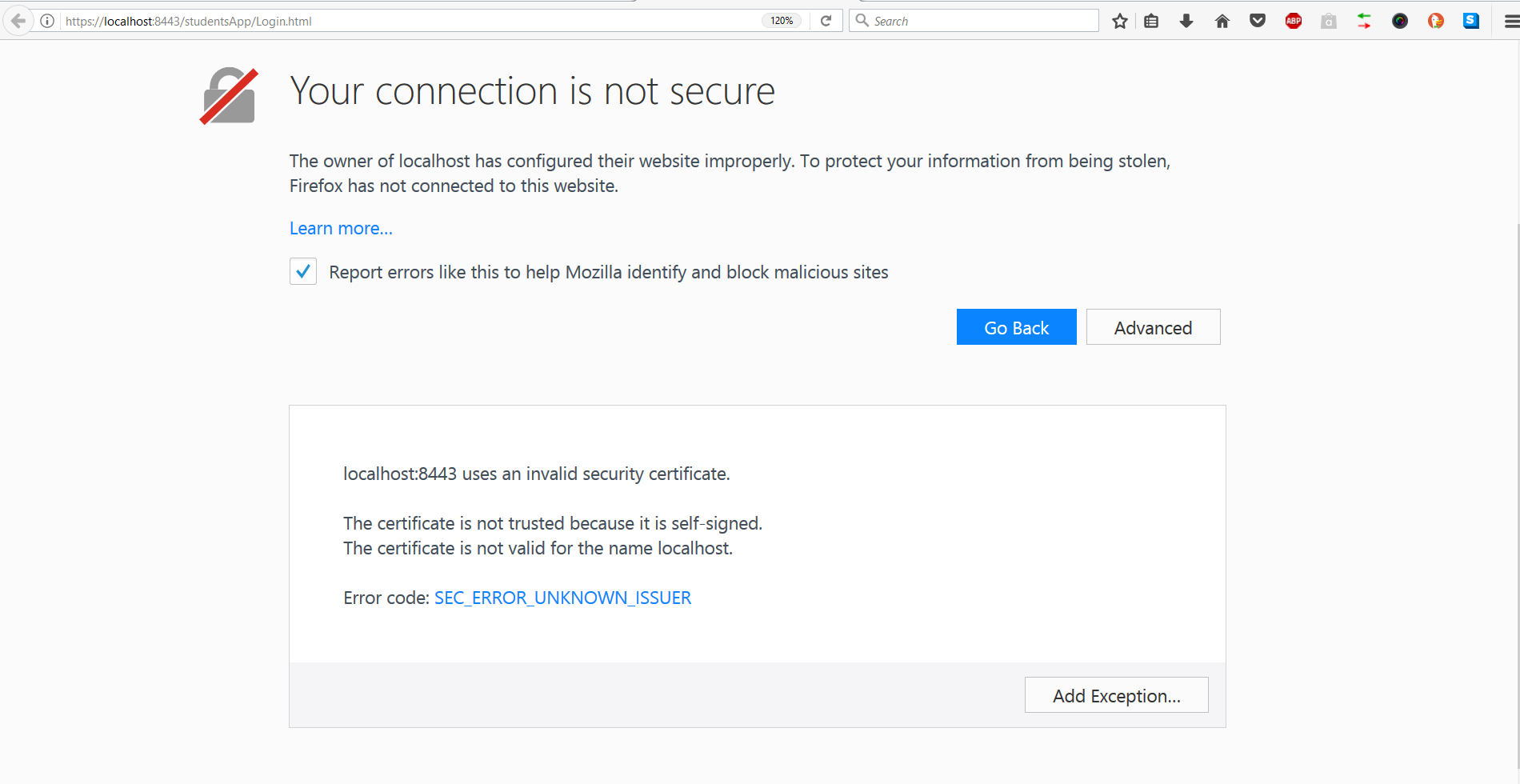
**Note**  
keystorePass="Admin@11" is the password you assigned to your keystore via “keytool” command.

1. **Restart the Tomcat and access using access to** <https://localhost:8443/>

**Note**: If you are using **Firefox:** You may notice a message will appear, this is caused by the self-signed certificate and Mozilla Firefox or Google chrome just do not trust it.

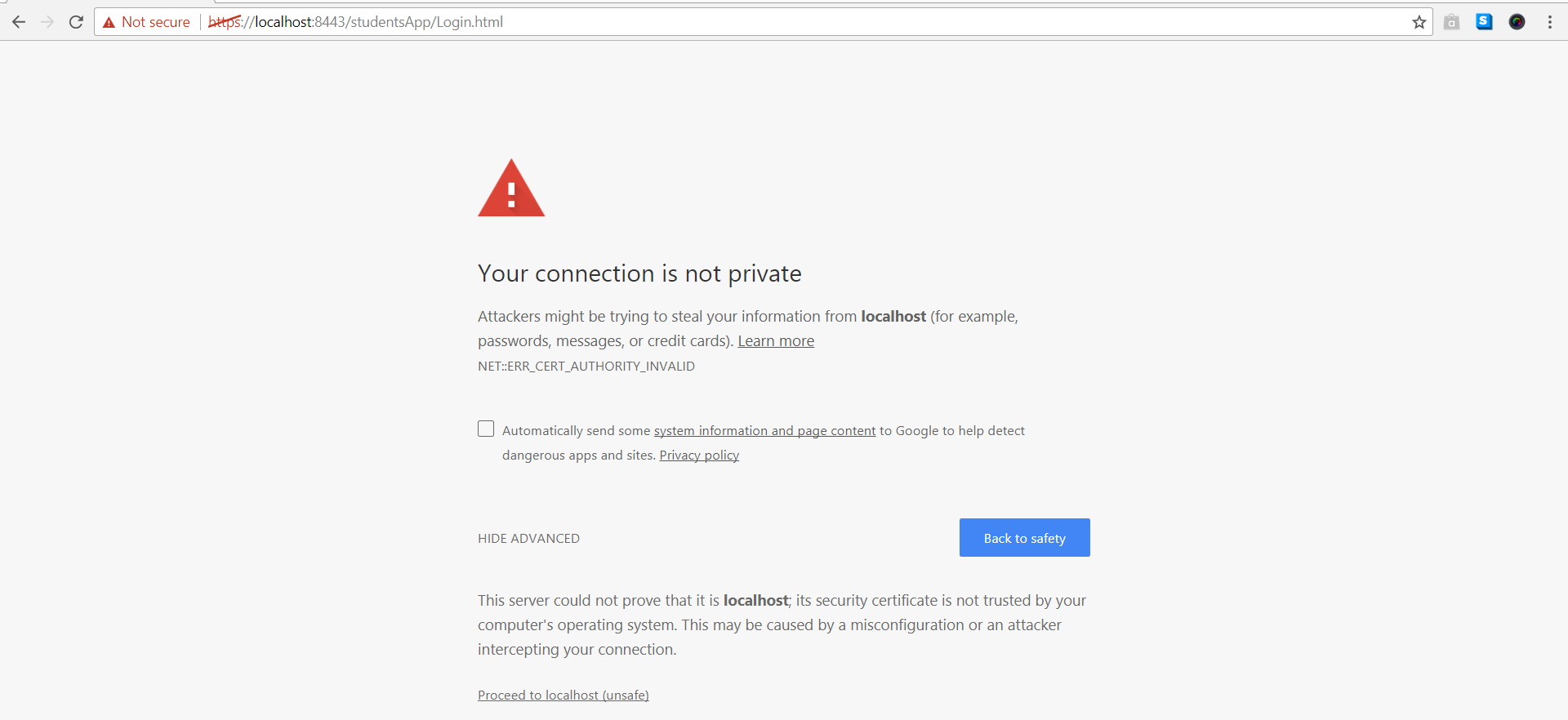
In production environment, you should consider buy a signed certificate from trusted SSL service provider like verisign or sign it with your own CA server

**Warning in Firefox**

****

Just say Add Exception and it will use the Https Protocol.

Warning in **Chrome**:



Just click on Proceed to <domain-name> and you’ll be able to see the browser using Https