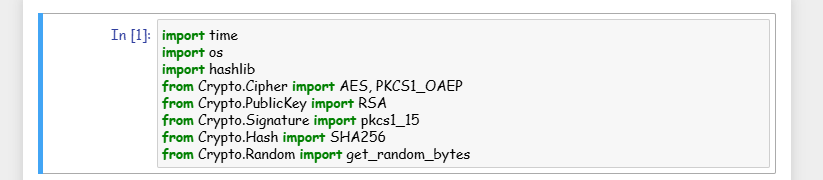
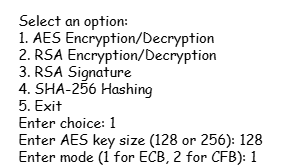
**Lab 4: Programming Symmetric & Asymmetric Crypto**

1. **Importing the necessary libraries:**

****

1. **AES encryption/decryption with two key lengths, 128 and 256 bits, and two modes ECB and CFB:**

****

**A close up of a text

Description automatically generated**

****

**A black text on a white background

Description automatically generated**

1. **RSA encryption and decryption:**

****

**A close-up of a computer code

Description automatically generated**

**A white background with black text

Description automatically generated**

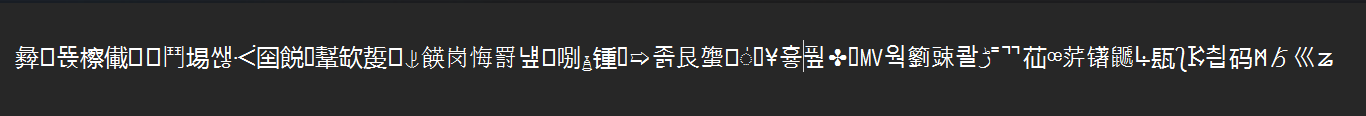
1. **RSA Signature:**

****

**A close-up of a document

Description automatically generated**

* **RSA signature:**

****

1. **SHA-256 hashing:**

**A screenshot of a computer

Description automatically generated**

**A black text on a white background

Description automatically generated**

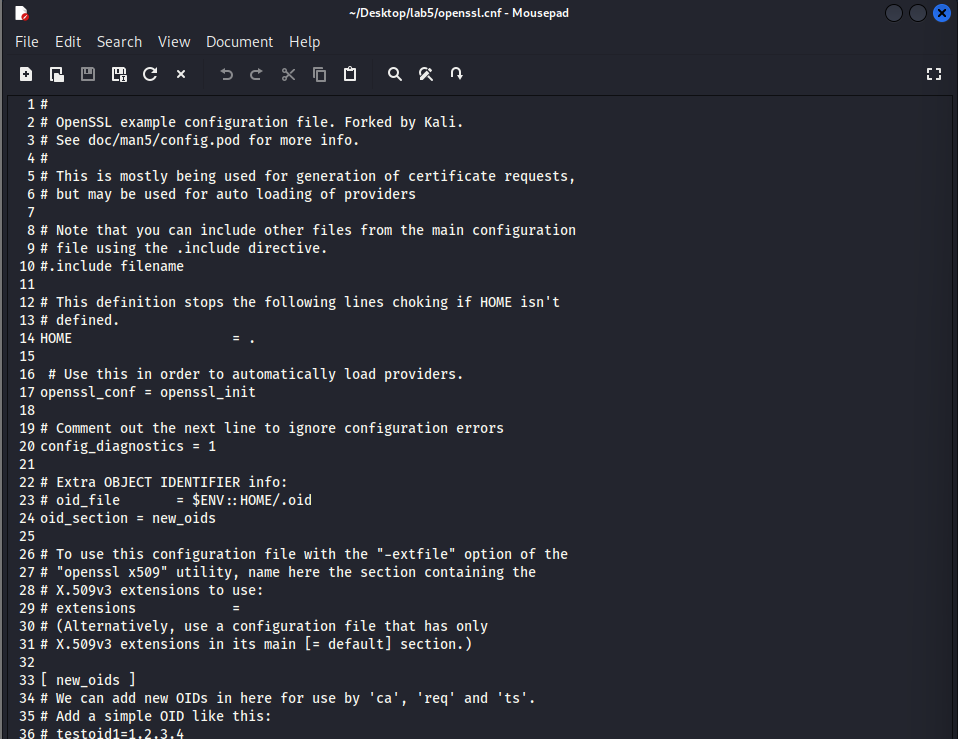
**Lab 5: Securing Apache Web Server**

**Task-1: Becoming a certificate authority**

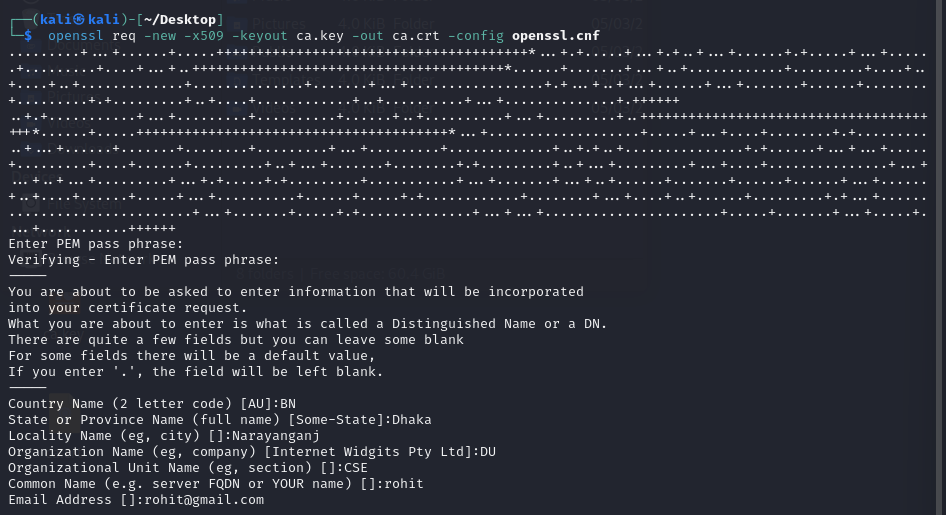
1. **Configure openssl.cnf:**
2. Copying the openssl.cnf file from /usr/lib/ssl/openssl.cnf to current directory.
3. Modifying openssl.cnf to suit the requirements, focusing on adjustments within the [CA\_default] section to define directories and settings appropriately.

**A black background with white text

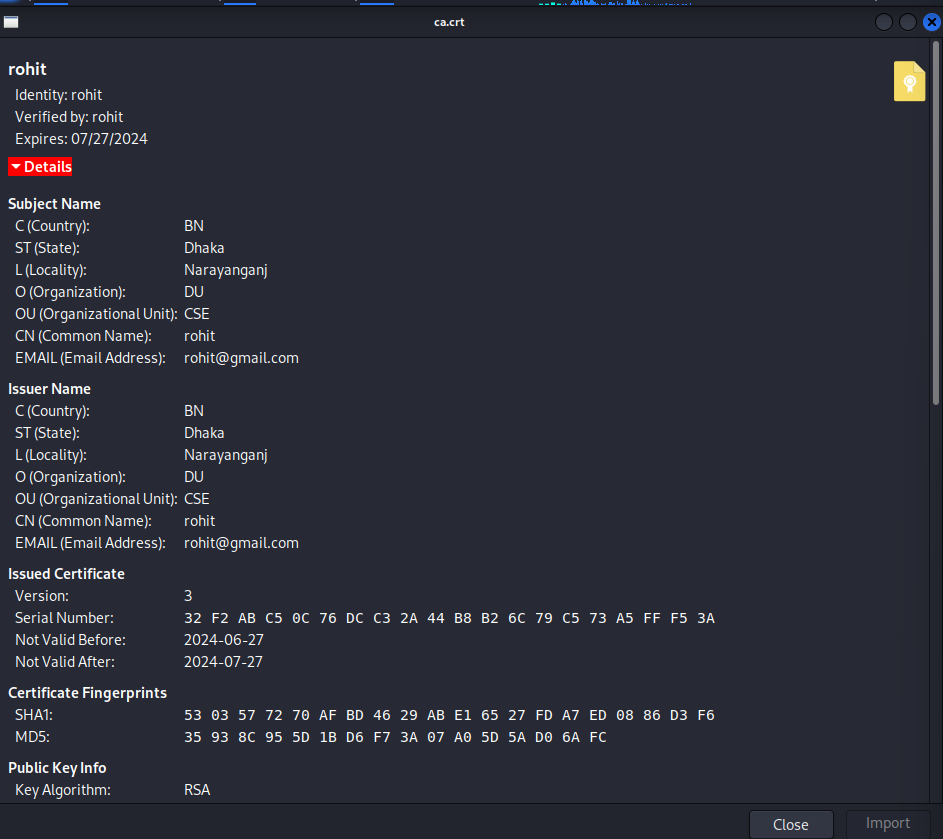
Description automatically generated**

****

1. **Generating a self-signed certificate for our CA, filling in necessary information such as Country Name, Common Name, etc.**

****

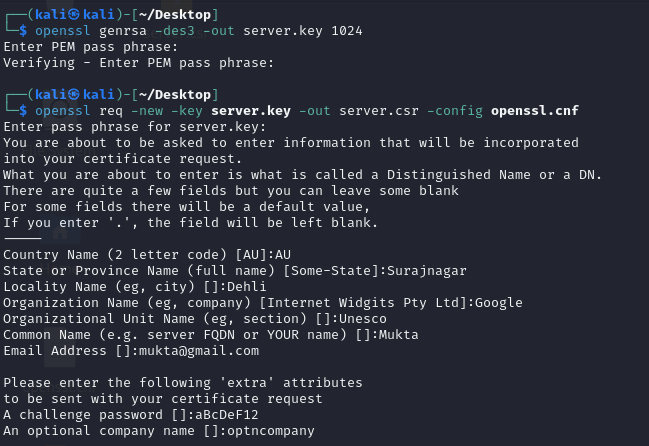
The output of the command is stored in two files: ca.key and ca.crt. The file ca.key contains the CA’s private key, while ca.crt contains the public-key certificate.

** A screenshot of a computer

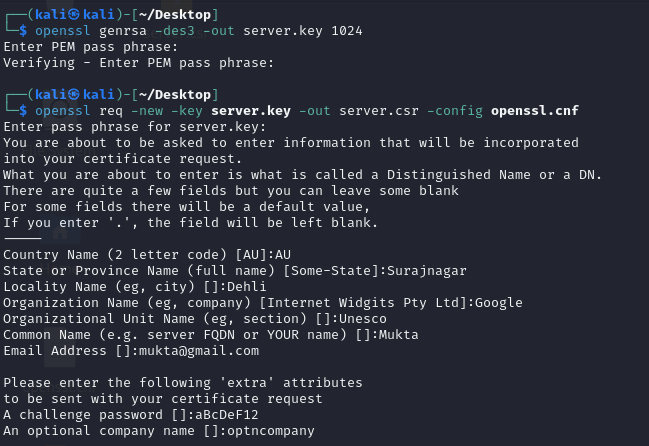
Description automatically generated**

**Task-2: Creating a certificate for example.com**

1. **Generating public/private key pair.**

****

1. **Generating a Certificate Signing Request (CSR).**

****

1. **Generating Certificates**
2. The names in your requests do not match with those of CA. So, OpenSSL refuses to generate certificates.

A screen shot of a computer code

Description automatically generated

1. Fix this and re-issue the above command.

**A screenshot of a computer screen

Description automatically generated**

**A computer screen shot of a computer

Description automatically generated**

**Task-3: Launching a simple web server with the certificate generated.**

1. **Combining the secret key and certificate into one file.**

**A screen shot of a computer program

Description automatically generated**

1. **Launch the web server using server.pem.**

**A screen shot of a computer program

Description automatically generated**

****

1. **error message from the browser**

**A screenshot of a computer

Description automatically generated**

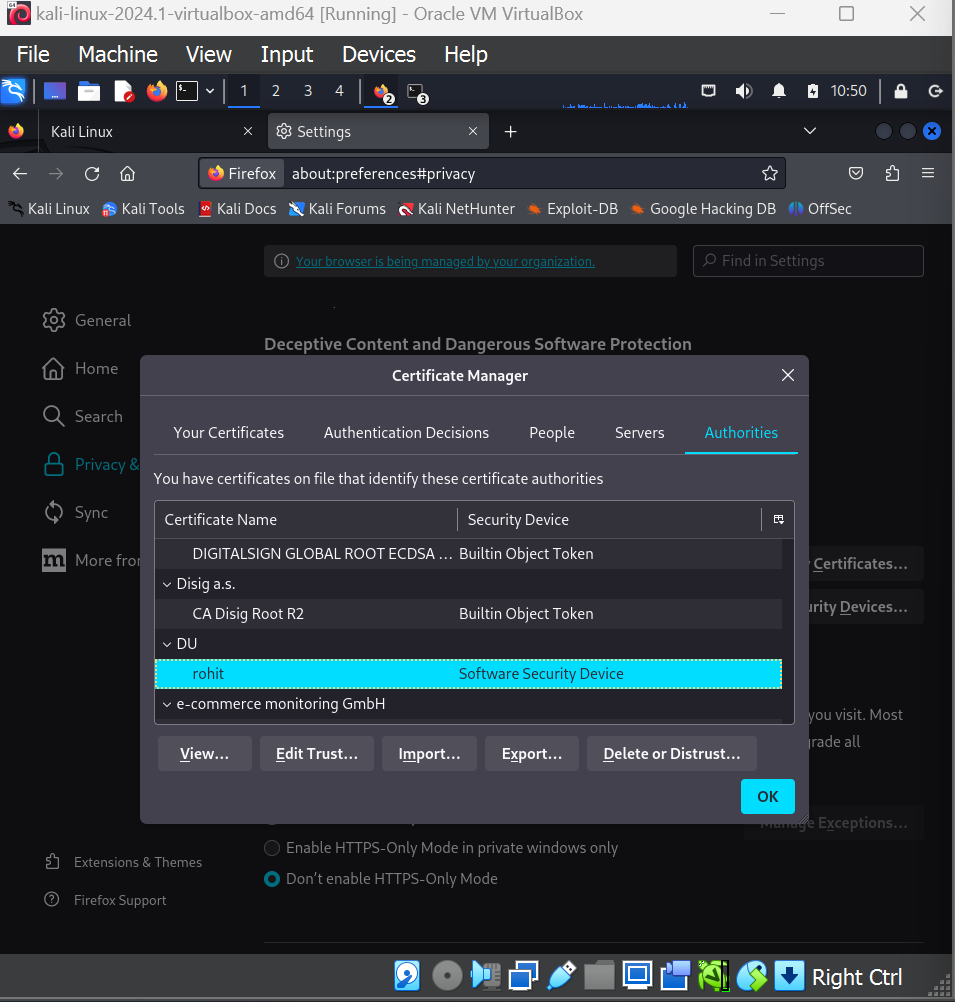
1. **Manually adding our CA’s certificate to the Firefox browser. A screenshot of a computer

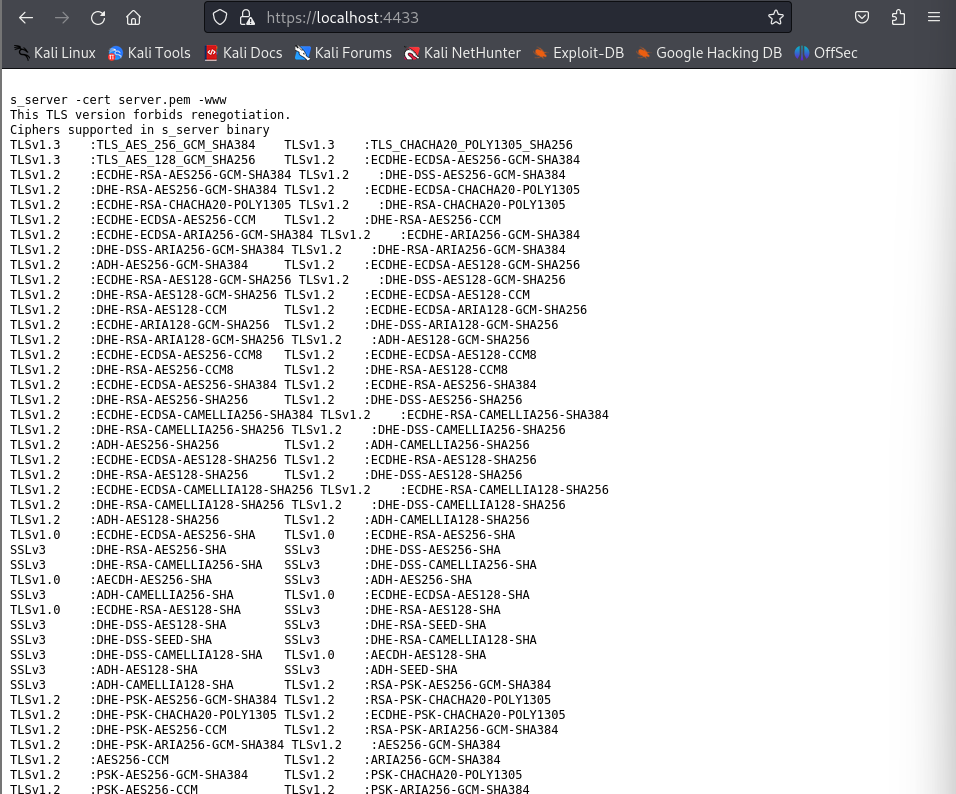
   Description automatically generated A screenshot of a computer

   Description automatically generated A screenshot of a computer

   Description automatically generated A screenshot of a computer

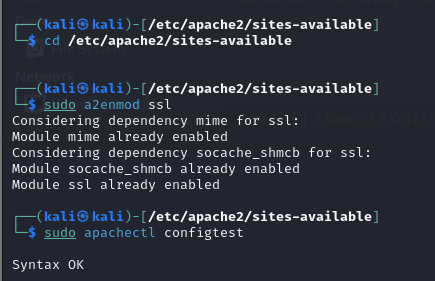
   Description automatically generated A screenshot of a computer

   Description automatically generated **
2. **In webpage, showing certificaticates’ details.**

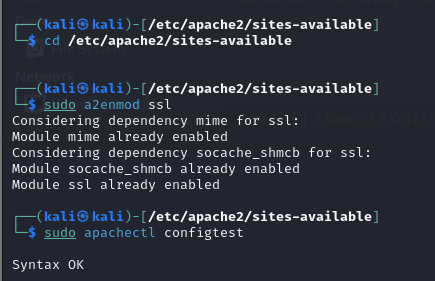
****

**Task-4: Deploy HTTPS into Apache**

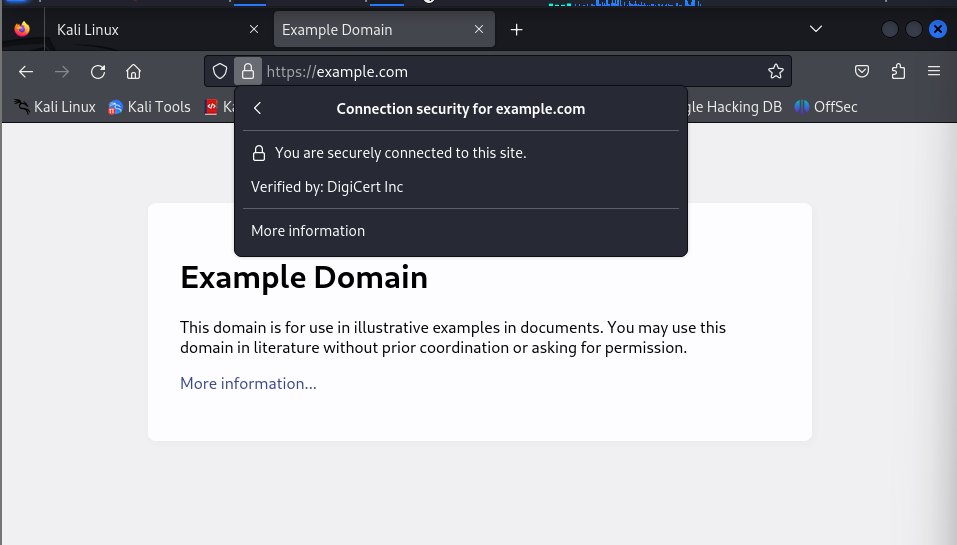
1. **Writing contents in /etc/apache2/sites-available/example.com.conf file.**

****

**A screenshot of a computer

Description automatically generated **

1. **Restarting the apache server.**
2. **Now, try to access the** [**http://example.com**](http://example.com)**. It’ll view the webpage in HTTPS.**

****

**Lab 6: Securing Apache Web Server – 2**

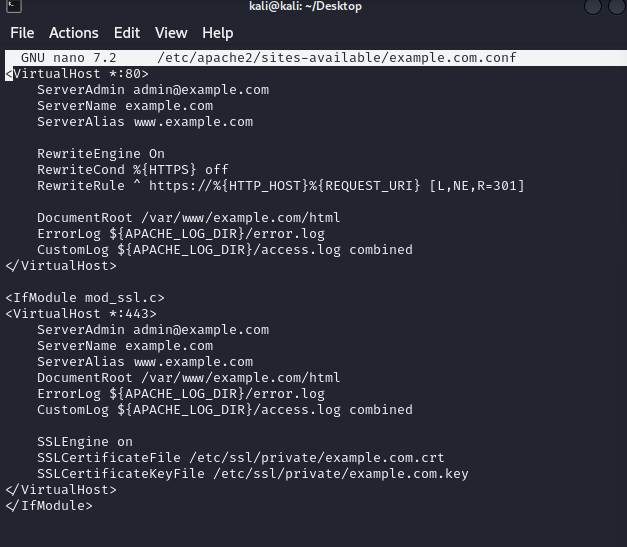
**Task – 1:**

1. **Enable the mod\_rewrite module using a2enmod command.**

**A screen shot of a computer

Description automatically generated**

1. **Look at the /etc/apache2/sites-enabled directory to find the configuration file for port 80 for example.com**

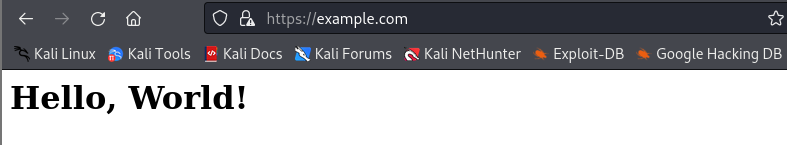
****

1. **Testing Apache configuration and restart the Apache server.**

**A screen shot of a computer

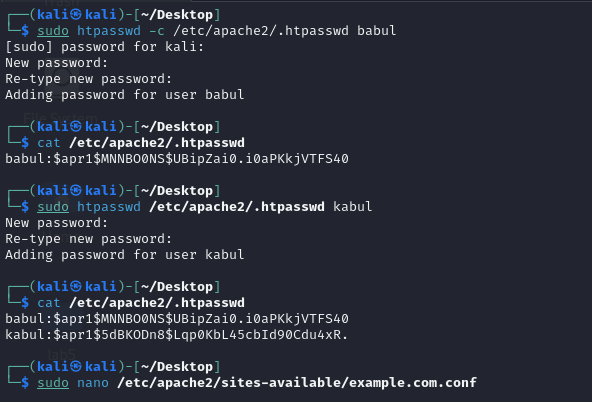
Description automatically generated**

1. **Testing example.com on your browser. It’ll show “Hello, World!”**

****

**Task-2:**

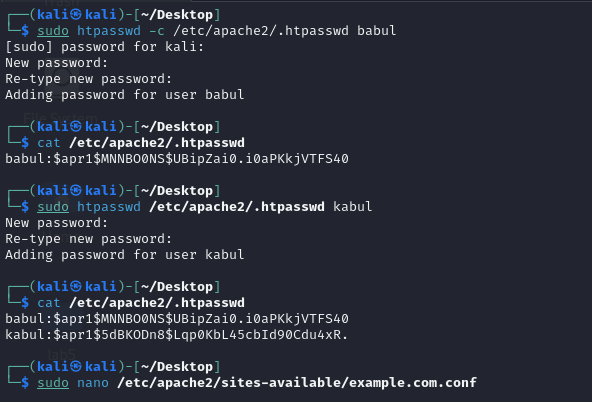
1. **Adding users to Apache web server using the following command:**

****

**A computer screen shot of a program

Description automatically generated**

1. **Use the following command to cat the contents of .htpasswd file.**

****

1. **Adding a few lines into https configuration file for example.com.**

**A screenshot of a computer

Description automatically generated**



1. **Restarting the apache server.**
2. **When accessing to example.com, it’ll show a prompt to give username and password.**

**A screenshot of a computer

Description automatically generated**

1. **Now providing the username and password from before created.**

**A screenshot of a computer

Description automatically generated**

1. **Now, I can access the site.**

**A screenshot of a computer

Description automatically generated A computer screen shot of a person

Description automatically generated**

**Task-3:**

1. **Installing and configuring Mariadb server on kali linux.**

**A computer screen shot of a computer code

Description automatically generated A screen shot of a computer

Description automatically generated**

1. **Creating a database called ‘apache’ and use the following command to use the apache database.**

**A screen shot of a computer

Description automatically generatedA screen shot of a computer code

Description automatically generated**

1. **Creating a table called users**

**A screen shot of a computer code

Description automatically generated**

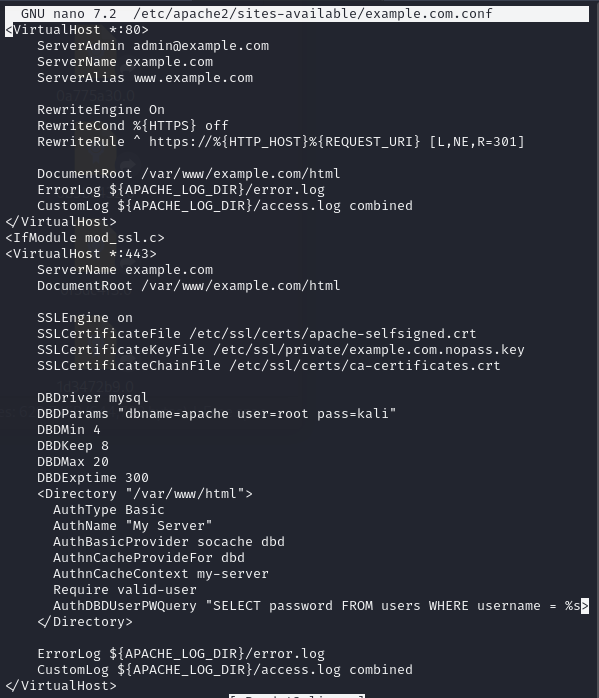
1. **Instead of storing plain passwords in database, use the following command to create a hashed password for users named Sammy and Alice in a separate consoleA screenshot of a computer program

   Description automatically generated**
2. **Now store the hashed password**

**A screenshot of a computer screen

Description automatically generated**

1. **Enable the mod\_authn\_dbd module of Apache and modifying configuration file’s contents.**

****

1. **Restart the Apache server.**
2. **When accessing the example.com page, it’ll show a prompt for username/password. Providing the one that have in your MySQL database. Now you can access the page.**

**A screenshot of a computer

Description automatically generated**