IERG 4210
Web Programming & Security
Tutorial 5

CHEN Lin

(Modified from the slides of former TA William Mui)

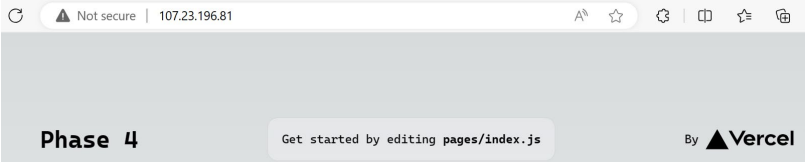
Previous Phase

- Main Website
  - Phase 2 - Content dynamically from database
  - Phase 3 - AJAX shopping list
- Admin panel
  - Phase 2 - Maintain the product database
- AWS EC2 Server
  - Phase 2 – your website is accessible at http://[your-own-public-IP]

Domain Name

- sxx.ierg4210.ie.cuhk.edu.hk
- secure.sxx.ierg4210.ie.cuhk.edu.hk
- Do not release your elastic IP!

http://[your-own-public-IP]



http://sxx.ierg4210.ie.cuhk.edu.hk

http://secure.sxx.ierg4210.ie.cuhk.edu.hk

default page of Nginx

Domain Name

- Configure the VM so that your website is accessible at http://[your-own-public-IP] and http://sxx.ierg4210.ie.cuhk.edu.hk
- Similar to Phase 2 Nginx Configuration (in Tutorial 2 Page 16)
  - Edit the Nginx configuration file with `nano`

    ```
    nano /etc/nginx/sites-available/nextjs.conf
    ```
  - Add your domain name in nextjs.conf

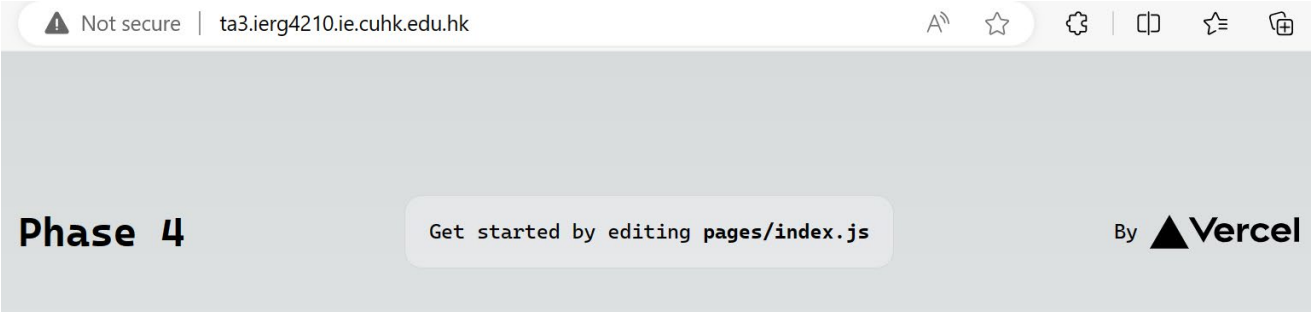
    ```
    server {
    listen 80;
    server_name  107.23.196.81   # Replace to your elastic IP
                 sxx.ierg4210.ie.cuhk.edu.hk; # Replace to your  domain name
    ```
  - Then press `Ctrl+O` to save modification, hit `Enter`, and press `Ctrl+X` to exit nano editor.
  - Test configuration file. If ok, reload Nginx

    ```
    sudo nginx -t
    sudo systemctl reload nginx
    ```

    ```
    ubuntu@ip-172-31-87-112:~$ sudo nginx -t
    nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
    nginx: configuration file /etc/nginx/nginx.conf test is successful
    ubuntu@ip-172-31-87-112:~$ sudo systemctl reload nginx
    ```

Domain Name

- Your website is accessible at http://[your-own-public-IP] and http://sxx.ierg4210.ie.cuhk.edu.hk (Not Secure)

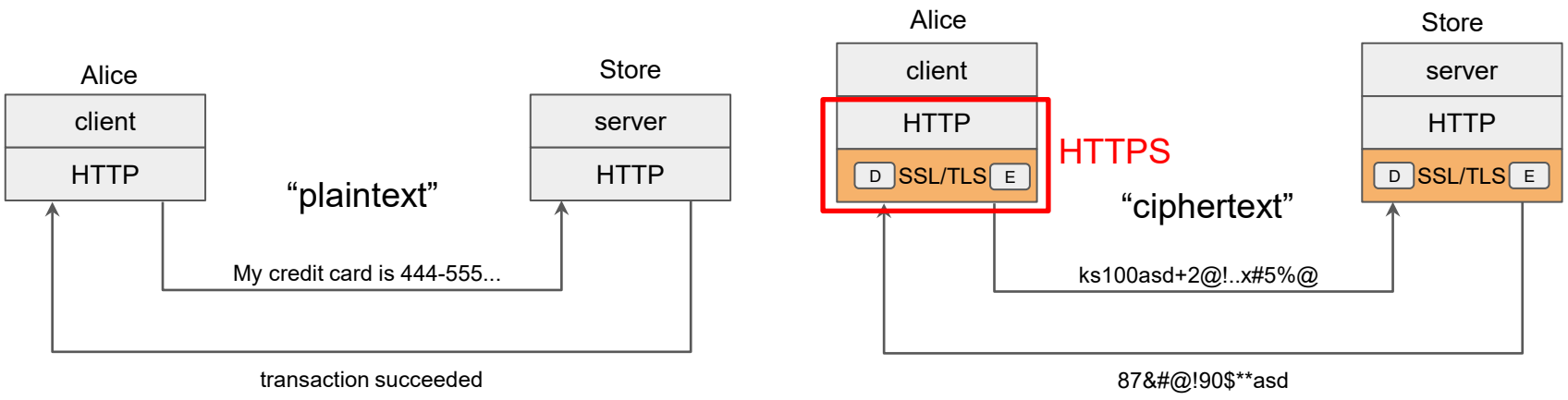


- Next: Secure your website https://secure.sxx.ierg4210.ie.cuhk.edu.hk

Phase 4: Secure your website

- Prevent XSS, CSRF, SQL attacks (Phase 4.1-4.3, 4.5)
- **Authentication for Admin Panel (Phase 4.4, 4.5)**
  - Otherwise everyone can manipulate your database.
- **Apply SSL certificate (Phase 4.6)**
  - Do it first, it takes time to apply

TLS/SSL Transport Layer Security / Secure Sockets Layer



- To make the whole procedure "secure":
  - Alice's credit card information can not be eavesdropped during the transition.
  - Credentiality ⇒ Symmetric Encryption and Decryption
- Alice's credit cart information can not be tampered during the transition.
  - Integrity ⇒ Message Authentication Code

Apply TLS/SSL to your website

Replace it with your number here!

- Create RSA Private Key and Certificate Signing Request (CSR)

- In your shell, input following commands
- `openssl req -nodes -newkey rsa:2048 -keyout secure.sxx.ierg4210.ie.cuhk.edu.hk.key -out server.csr`

  - `openssl` - activates the OpenSSL software.
  - `req` - indicates that we want a CSR.
  - `-new -newkey` - generates a new key.
  - `rsa:2048` - generates a 2048-bit RSA mathematical key.
  - `-nodes` - no DES, meaning do not encrypt the private key in a PKCS#12 file.
  - `-keyout` - indicates the domain for which you are generating a key.
  - **-out** - specifies the name for saving the CSR file.

Apply TLS/SSL to your website

Replace it with your number here!

- ssh to your server
- In your shell, input following commands to create **RSA Private Key** and **Certificate Signing Request (CSR)**
- `openssl req -nodes -newkey rsa:2048 -keyout secure.sxx.ierg4210.ie.cuhk.edu.hk.key -out server.csr`
- In the interactive prompt:
  - Country Name (2 letter code) [XX]:HK
  - State or Province Name (full name) []:Hong Kong
  - Locality Name (eg, city) [Default City]:
  - Organization Name (eg, company) [Default Company Ltd]:CUHK
  - Organizational Unit Name (eg, section) []:
  - Common Name (eg, your name or your server's hostname) []:secure.sxx.ierg4210.ie.cuhk.edu.hk
  - Email Address []:your email
- DO NOT input password at this step or your servercan not read it!

Apply TLS/SSL to your website

```
ubuntu@ip-172-31-87-112:~$ openssl req -nodes -newkey rsa:2048 -keyout secure.ta3.ierg42
10.ie.cuhk.edu.hk.key -out server.csr
```

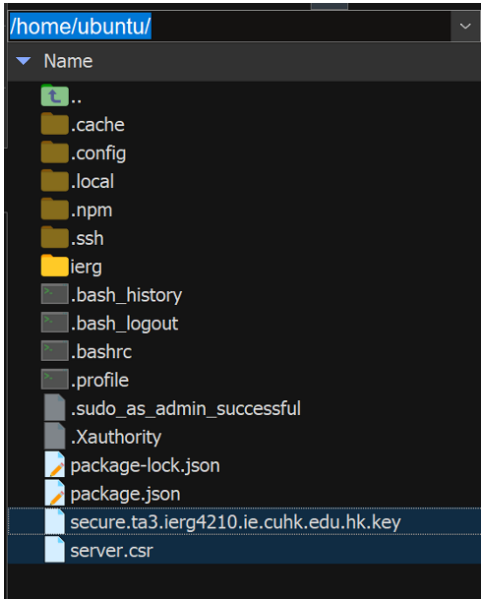
```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:HK
State or Province Name (full name) [Some-State]:Hong Kong
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CUHK
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:secure.ta3.ierg4210.ie.cuhk.edu.hk
Email Address []:cl022@ie.cuhk.edu.hk
```

- DO NOT input password at this step or your servercan not read it!

```
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
ubuntu@ip-172-31-87-112:~$
```

Apply TLS/SSL to your website

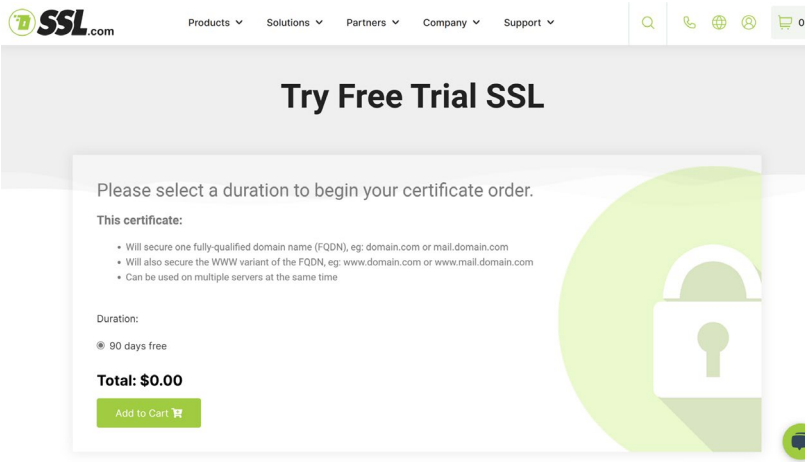
- **Just put the csr and key file in somewhere inaccessible by common users**
- `cat server.csr`

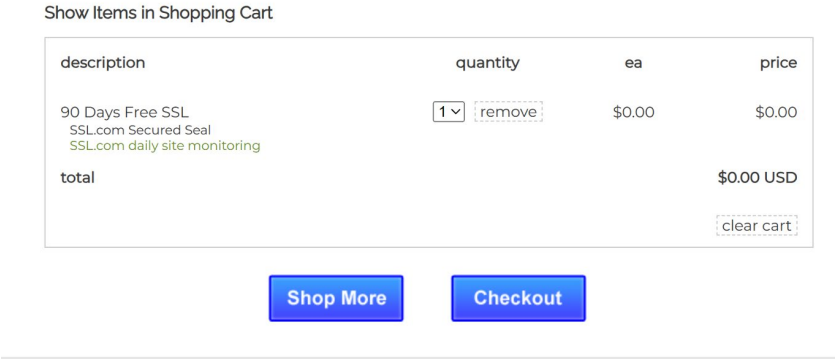


```
ubuntu@ip-172-31-87-112:~$ cat server.csr
-----BEGIN CERTIFICATE REQUEST-----
MIICzzCCAbcCAQAwgYkxCzAJBgNVBAYTAkhLMRIwEAYDVQQIDAlIb25nIEtvbmcx
DTALBgNVBAoMBENVSEsxKzApBgNVBAMMInNlY3VyZS50YTMuaWVyZzQyMTAuaWUu
Y3Voay5lZHUuaGsxKjAoBgkqhkiG9w0BCQEWG2xpbmNoZW4xNTI4NzE4Nzg0QGdt
YWlsLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANRDWqgBeJx7
dXnYY1jsjmC75fOJm1Mdu/g4Ieti2Rp6mea0isCsT97mBJFwz5EC6ESzbgEmHt03
r24lIVjewUUxW8KIOuzjh7ZF2qF48oamkcqKZbSZtZ7AydcTwg0X9agH9RAlvmxk
KOeEIw1dnyMxFsZuMKz4qIaLFVPRvyjloK9IoXkhhHXGYQa7NzYOrS74usq5Vmvy
7XMCiKjzcjnep9oTr9boIvQyHA9slt1V7Rh9TKPHPBatstUPDearPCC75e63gxaS
Fhvvx292LY3P6sRBdeMC8HgomfX75RMO/xhAZwsIdyNMCRRJiCVWfZNnUTQ5US6k
L3ejmM4fUvECAwEAAaAAMA0GCSqGSIb3DQEBCwUAA4IBAQAebInM/Hpj6GLKR22E
Hqb2jS4TEU8B6kXKgy8hi0WhO2cqn8vFOnJvfkluFFI9xsTIAC7P1MOprunuoRTJ
q8kIyoWmNfYcp/UujeCLO3i9BRU5IDBtPAfmHd4YLDYgRHhePB3X6XXH6xwQFjsm
W23XVXUsa/f2FM+YFWtMMr6ZDNrV145GybYwPTvXUqz5UgfKejGgqM0J1AiKHYo+
3I/8yBv24H/OPVtU7teusZL7spKSBxpfdH8Lq74MZS2g+Mg1CRfR2MmyIgxchWov
A6zS5cWJyzsubedFUsY5pS/aU/buFd/s90kAvQu8n2b4817XM+qxf52Gqr6fwlLb
yXPg
-----END CERTIFICATE REQUEST-----
ubuntu@ip-172-31-87-112:~$
```

Sign up for free

- https://www.ssl.com/certificates/free/buy is illustrated here





Order successfully placed. Click here to finish processing your ssl.com certificates.

Show Order Transaction

Click here to finish processing this certificate order.

Order Details

reference number: 3bc3-1iujh2t
date of order: March 07, 2024
payment method: n/a (free)

DESCRIPTION	QUANTITY	PRICE
90 Days Free SSL SSL.com Secured Seal SSL.com daily site monitoring co-dc1iujh2t0c	1	

Subtotal: $0.00

Tags

click here to add/remove/edit searchable tags f

Subject	Team Folder Expires	Reference#	Status	Order Date	Action
90 Days Free SSL certificate open : finish processing	ad5-1iujh2r n/a	co-dc1iujh2t0c	waiting for csr	Mar 07, 2024	submit csr

10 Export to CSV

Note: The Common Name (CN) field in your CSR must conform to the following rules:

- It MUST represent your fully-qualified domain name (i.e. submit.example.com)

*CSR:
DO NOT SUBMIT YOUR PRIVATE KEY!


```
MIICzzCCAbcCAQAwgYkxCzAJBgNVBAYTAkhLMRIwEAYDVQQ
DTALBgNVBAoMBENVSEsxKzApBgNVBAMMInNlY3VyZS50YTM
Y3Voay5lZHUuaGsxKjAoBgkqhkiG9w0BCQEWG2xpbmNoZW4
YW1sLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQc
dXnYY1jsjmC75fOJm1Mdu/g4Ieti2Rp6meaOisCsT97mBJF
r24lIVjewUUxW8KIOuzjh7ZF2qF48oamkcqKZbSZtZ7Aydc
KOeEIw1dnyMxFsZuMKz4qIaLFVPRvyjloK9IoXkhhHXGYQa
7XMCiKjzcjnep9oTr9boIvQyHA9slt1V7Rh9TKHPBatstU
Fhvvx292LY3P6sRBdeMC8HgomfX75RMO/xhAZwsIdyNMCRR
```


Save to CSR Manager: ☑

Managed CSR: none Create CSR

Common Name (CN):
primary domain name

Server Software:
for informational purposes only: OTHER

Schedule SSL Scans:
expiration reminder notifications: ◉ Simple ○ Custom ○ None

Daily (at midnight)

Delegated Credentials: ☐

*Subscriber Agreement ☑ By clicking this check box, you agree to the terms of the SSL.com Subscriber Agreement

- **Open server.csr you created and paste into the field**

  `cat server.csr`

- **Start with**

  -----BEGIN CERTIFICATE REQUEST-----


```
ubuntu@ip-172-31-87-112:~$ cat server.csr
-----BEGIN CERTIFICATE REQUEST-----
MIICzzCCAbcCAQAwgYkxCzAJBgNVBAYTAkhLMRIwEAYDVQQIDAlIb25nIEtvbmcx
DTALBgNVBAoMBENVSEsxKzApBgNVBAMMInNlY3VyZS50YTMuaWVyZzQyMTAuaWUu
Y3Voay5lZHUuaGsxKjAoBgkqhkiG9w0BCQEWG2xpbmNoZW4xNTI4NzE4Nzg0QGdt
YW1sLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANRDWqgBeJx7
dXnYY1jsjmC75fOJm1Mdu/g4Ieti2Rp6meaOisCsT97mBJFwz5EC6ESzbgEmHt03
r24lIVjewUUxW8KIOuzjh7ZF2qF48oamkcqKZbSZtZ7AydcTwg0X9agH9RAlvmxk
KOeEIw1dnyMxFsZuMKz4qIaLFVPRvyjloK9IoXkhhHXGYQa7NzYOrS74usq5Vmvy
7XMCiKjzcjnep9oTr9boIvQyHA9slt1V7Rh9TKPHPBatstUPDearPCC75e63gxaS
Fhvvx292LY3P6sRBdeMC8HgomfX75RMO/xhAZwsIdyNMCRRJiCVWfZNnUTQ5US6k
L3ejmM4fUvECAwEAAaAAMA0GCSqGSIb3DQEBCwUAA4IBAQAebInM/Hpj6GLKR22E
Hqb2jS4TEU8B6kXKgy8hi0WhO2cqn8vFOnJvfkluFFI9xsTIAC7P1MOprunuoRTJ
q8kIyoWmNfYcp/UujeCLO3i9BRU5IDBtPAfmHd4YLDYgRHhePB3X6XXH6xwQFjsm
W23XVXUsa/f2FM+YFWtMMr6ZDNrV145GybYwPTvXUqz5UgfKejGgqM0J1AiKHYo+
3I/8yBv24H/OPVtU7teusZL7spKSBxpfdH8Lq74MZS2g+Mg1CRfR2MmyIgxchWov
A6zS5cWJyzsubedFUsY5pS/aU/buFd/s9OkAvQu8n2b4817XM+qxf52Gqr6fwlLb
yXPg
-----END CERTIFICATE REQUEST-----
ubuntu@ip-172-31-87-112:~$
```

Validation

will become green if you pass the validation

Remove Edit	secure.ta3.ierg4210.ie.cuhk.edu.hk	CSR hash text file usinc	failed	validation not performed yet	n/a	waiting
	Instructions: Upload THIS FILE to this location: http://secure.ta3.ierg4210.ie.cuhk.edu.hk/.well-known/pki-validation/060508AFBF0A774D9CD865E3CC7F7033.txt					

Select “CSR hash text file using http://”.

Do not use ‘email’ as validation method, otherwise the IE admin will receive many emails

- Follow the instruction to download the xxx.txt file
- Upload the xxx.txt file to AWS EC2 server
- Copy xxx.txt file to a dedicated folder /.well-known/pki-validation

```
sudo mkdir -p /usr/share/nginx/html/.well-known/pki-validation
sudo cp xxx.txt /usr/share/nginx/html/.well-known/pki-validation
```

Validation

Remove Edit	secure.ta3.ierg4210.ie.cuhk.edu.hk	CSR hash text file usinc	failed	validation not performed yet	n/a	waiting

Instructions: Upload THIS FILE to this location: http://secure.ta3.ierg4210.ie.cuhk.edu.hk/.well-known/pki-validation/060508AFBF0A774D9CD865E3CC7F7033.txt

- Edit the Nginx configuration file and add a new server block

```
nano /etc/nginx/sites-available/nextjs.conf
```

```
server {
listen 80;
server_name  secure.ta3.ierg4210.ie.cuhk.edu.hk; # Replace to your domain name

location ~ /.well-known {
root /usr/share/nginx/html;
allow all;
}

}
```

the path to /.well-known/pki-validation

```
sudo systemctl reload nginx
```

不安全 | secure.ta3.ierg4210.ie.cuhk.edu.hk/.well-known/pki-validation/060508AFBF0...

```
1A6F6B333EBA9A43D0F466D6F1864851BD0B9F9FE91B46D39F56B189B7BD3C07
ssl.com
20240307
```

Validation

Select “CSR hash text file using http://”. Click “Validation”

Remove Edit	secure.ta3.ierg4210.ie.cuhk.edu.hk	CSR hash text file usinc	failed	validation not performed yet	n/a	waiting

Instructions: Upload THIS FILE to this location: http://secure.ta3.ierg4210.ie.cuhk.edu.hk/.well-known/pki-validation/060508AFBF0A774D9CD865E3CC7F7033.txt

will become green if you pass the validation

Monitoring

Free Certificate For secure.ta3.ierg4210.ie.cuhk.edu.hk [how do I use this page?]

Subject	Team	Reference#	Folder	Status	Order Date	Expires	Action
secure.ta3.ierg4210.ie.cuhk.edu.hk open : download : documents : seal	ad5-1iujh2r	co-dc1iujh2t0c		issued	Mar 07, 2024	Jun 06, 2024	[upgrade] [change domain(s)/rekey]

certificate details		validation status		smart seal
certificate type	duration	validation level	issued on	requested on
Free	90 days	Class 1 DoD	Mar 08, 2024	Mar 08, 2024

Download the CRL File

CERTIFICATE DETAILS

^ Hide Details

certificate contents
algorithm: sha256WithRSAEncryption

validation documentation
none submitted
upload | status

verify and troubleshoot
check ssl installation
visit site with ssl
visit site without ssl

for developers
preformatted api strings
developer tools
how to use ACME

registrant
none

subject dn
CN=secure.ta3.ierg4210.ie.cuhk.edu.hk

Notification Groups
ng-co-dc1iujh2t0c

INVITE USERS

certificate downloads

Microsoft IIS (*.p7b)	download	guide
WHM/cpanel	download	guide
Apache	download	guide
Amazon	download	guide
Nginx	download	guide
V8+Node.js	download	guide
Java/Tomcat	download	guide
Other platforms	download	guide
CA bundle (intermediate certs)	download	guide

use ACME with nginx and apache

Download Nginx file (.crt)
Follow the guide

/home/ubuntu/

Name
..
.cache
.config
.local
.npm
.ssh
.well-known
ierg
.bash_history
.bash_logout
.bashrc
.profile
.sudo_as_admin_successful
.Xauthority
package-lock.json
package.json
secure.ta3.ierg4210.ie.cuhk.edu.hk.key
secure_ta3_ierg4210_ie_cuhk_edu_hk.chained.crt
server.csr

Uploaded crt to the sever

Configure an HTTPS server

```
nano /etc/nginx/sites-available/nextjs.conf
```

- Add the following content the configuration file

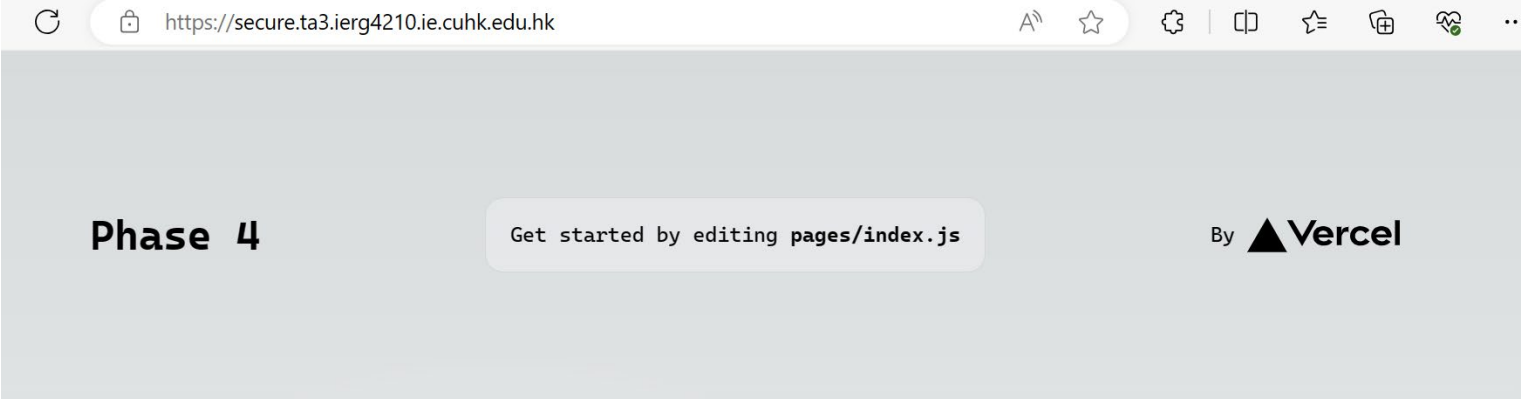
```
server {
listen 443 ssl;
server_name secure.sxx.ierg4210.ie.cuhk.edu.hk;
ssl_certificate /path/to/secure_sxx_ierg4210_ie_cuhk_edu_hk.chained.crt;
ssl_certificate_key /path/to/secure.sxx.ierg4210.ie.cuhk.edu.hk.key;

location / {
proxy_pass http://localhost:3000;
proxy_http_version 1.1;
proxy_set_header Upgrade $http_upgrade;
proxy_set_header Connection 'upgrade';
proxy_set_header Host $host;
proxy_cache_bypass $http_upgrade;
}
}
```

based on your domain name, your path to the server certificate chain file, and your path to private key file

Configure an HTTPS server (cont.)

- Open port 443 in AWS security group!
- Restart the server
  - `sudo nginx -t`
  - `sudo systemctl reload nginx`

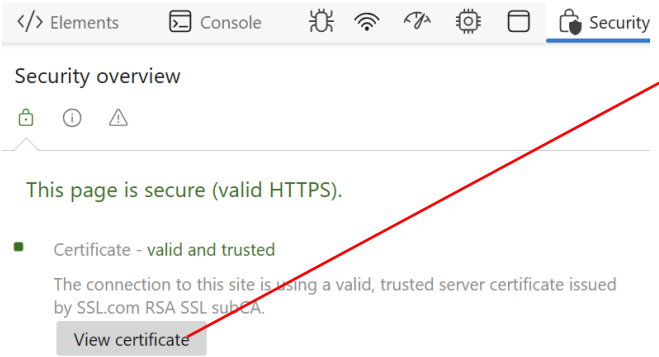
```
ubuntu@ip-172-31-87-112:~$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
ubuntu@ip-172-31-87-112:~$ sudo systemctl reload nginx
ubuntu@ip-172-31-87-112:~$
```

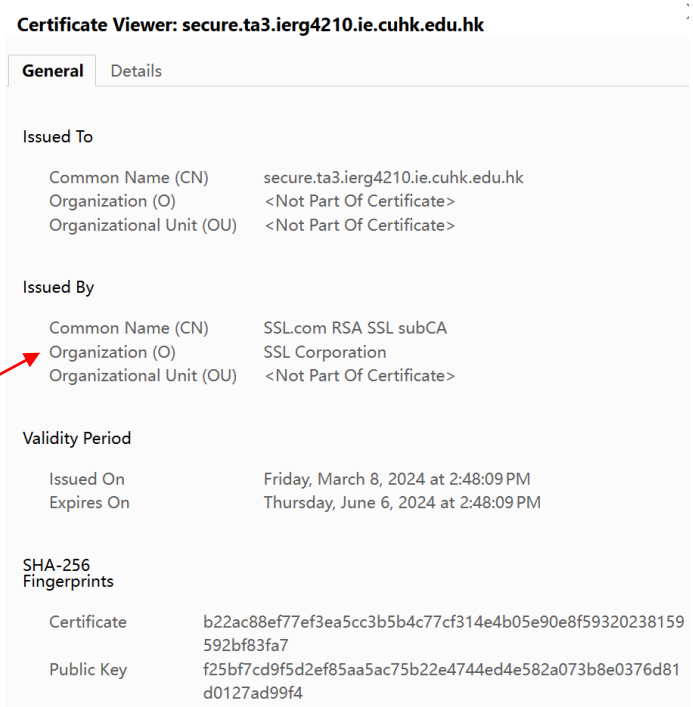


More configuration for an HTTPS server

Check the certificate

- Visit using the browser your website (https://secure.sxx.ierg4210.ie.cuhk.edu.hk)
- If you use Chrome
  - Developer Tool (F12)
  - Go to Security tab
  - View certificate





Redirect HTTP requests to HTTPS

- You can redirect user if they access

  http://secure.sxx.ierg4210.ie.cuhk.edu.hk -> https://secure.sxx.ierg4210.ie.cuhk.edu.hk

- Modify the configure file
  - `nano /etc/nginx/sites-available/nextjs.conf`

```
server {
listen 80;
server_name  secure.ta3.ierg4210.ie.cuhk.edu.hk; # Replace to your domain name

rewrite ^/(.*)$ https://secure.ta3.ierg4210.ie.cuhk.edu.hk/ permanent;

}
```

Redirect HTTP requests to HTTPS

- Restart the server
  - `sudo nginx -t`
  - `sudo systemctl reload nginx`

Authentication for Admin Panel – Phase 4.4

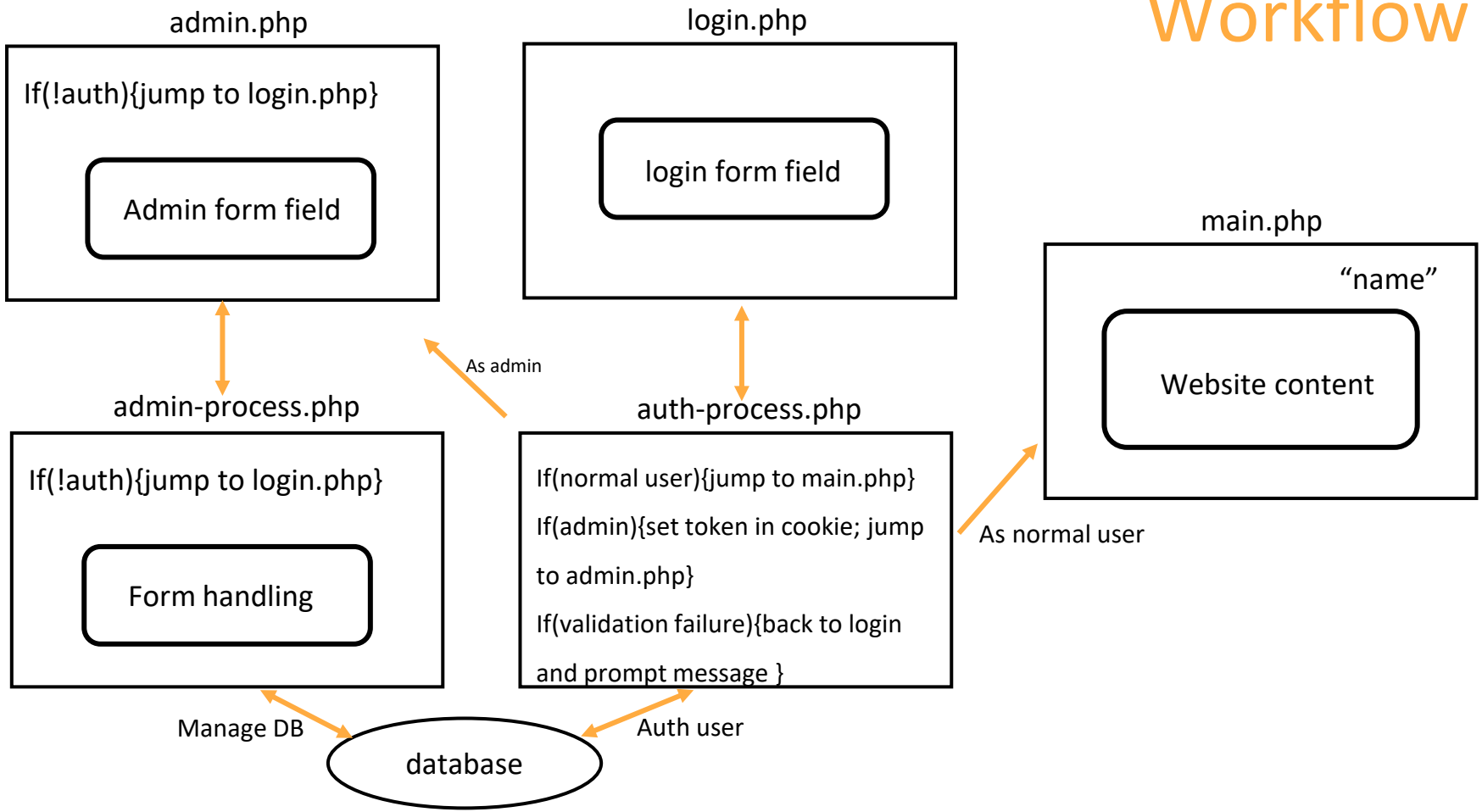
- A website page and an admin page. But everyone can access admin panel now.
  - We need to add an admin user to the user management database

    (only user with special privilege could visit admin page and do operations)
  - Store hashed passwords in database (user table) (// Why not original?)
  - Build a login page and perform the authentication.
  - Use cookies to remember the authentication result. (via maintaining the token)
  - Support logout and password changing

Phase 4.4

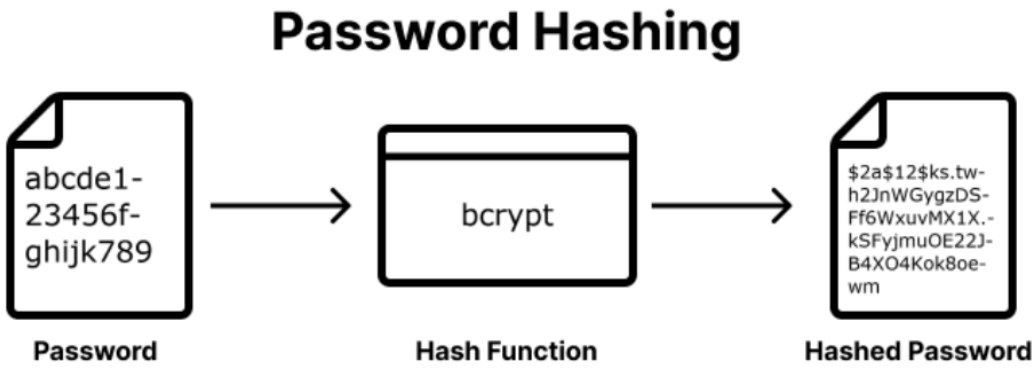
- Create a user table
- Login
- Maintain an authentication token using Cookies
- Validate the token
- Support logout and password changing

Workflow


admin.php
If(!auth){jump to login.php}
Admin form field
login.php
login form field
main.php
“name”
Website content
As admin
admin-process.php
If(!auth){jump to login.php}
Form handling
auth-process.php
If(normal user){jump to main.php}
If(admin){set token in cookie; jump to admin.php}
If(validation failure){back to login and prompt message }
As normal user
Manage DB
database
Auth user

Hashed password

- Saving user passwords in the database in plain text format is reckless. It is preferable to hash your password before storing it.
- For instance, it will be tough to decipher the passwords in your database if they are leaked. Hashing passwords is a cautious and reliable practice.



Hash Function

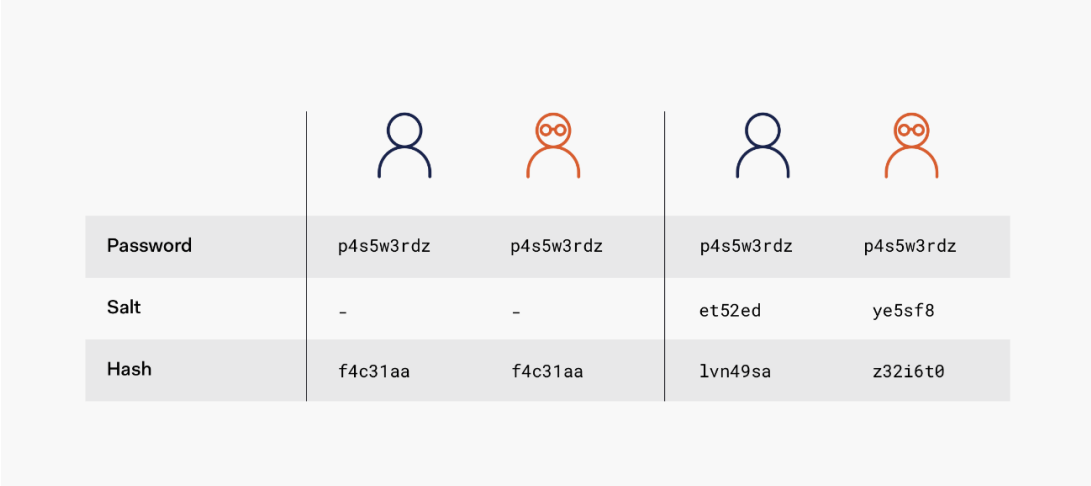
- Accept variable size message M and produce a fixed-size digest h(M)
  - h(M) can be thought as "fingerprint" of M
- A "good" hash function:
  - Easy to compute h(M)
  - Computationally infeasible to find M from h(M)
  - Computationally infeasible to find collision ($X \neq Y$, but h(X)=h(Y))

    However, collision always exists since the length of messages is longer than that of digest.

Password	p4s5w3rdz	p4s5w3rdz
Hash	f4c31aa	f4c31aa

Salted and hashed password

- Secure Hash Functions
  - Offline-dictionary attack: pre-computed a list of hashed values to create a lookup table
  - Salting, i.e., add a random string to expand the effective space for brute-force attack
  - Many hash functions, some are broken: MD5, SHA-1, SHA-256, …
  - Just call the existing libraries; don’t implement the algorithm yourself



Password	p4s5w3rdz	p4s5w3rdz	p4s5w3rdz	p4s5w3rdz
Salt	-	-	et52ed	ye5sf8
Hash	f4c31aa	f4c31aa	lvn49sa	z32i6t0

Database – User Table

- Create user table to save userid (primary key), email, salt, “salted **and** hashed password”, admin flag.
  - flag (e.g., integer 1) to indicate “admin” or not
- Every user has its own random salt, so the salted password generated by below will be different

```
<?php
echo ($salt = mt_rand()).“<br/>”;
echo hash_hmac('sha256', $_REQUEST['password'], $salt);
?>
```

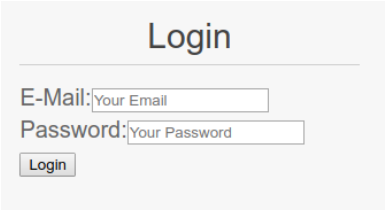
Message Authentication Code built from hash

Secret key for HMAC

- Adding a user: `INSERT INTO account (email, salt, password) VALUES`

```
("1@gmail.com","1160029811","5d2b3d93eba5eb05e34b7c2301c517a17c593bc364ca88fa3417944cb5a4e74d");
```

Login Page



- Build login.php and auth-process.php
  - Create the HTML yourself
  - Form will be submitted to auth-process.php
    - submit email, password (first validate the format)
    - get “salt” from DB, compute the “salted hash value” then compare.
    - lead admin to admin panel, common user to main page, refuse incorrect password.
- Now you have:
  - login, admin, mainpage
  - Related process file auth-process, admin-process
  - Every time need password?
    - Set admin token kept in cookie.

Reminder

- Watch out the Amazon billing notification
  - May charge you if you open redundant resources
- Secure your private key
- Backup your server data
- Domain names are released. Do not release your elastic IP!
- Do NOT hack your classmates' website at this stage!

Node.js User Authentication

1. https://www.loginradius.com/blog/engineering/guest-post/nodejs-authentication-guide/