

長庚大學工學院

學生校外實習期末報告

程式課程平台

Online Judge 部署流程文件

實習期間：自 112 年 7 月 3 日 至 112 年 8 月 25 日

輔導老師：張哲維 教授

學生：B0929260 張孟佳

致謝詞

感謝麥錫森智能科技股份有限公司賴昭榮博士之協助指導，以及專題指導老師張哲維教授之指導，組員之協調合作，並感謝家人朋友的關懷支持。

摘要

程式課程平台是我們結合觀看線上課程影片以及線上批改系統的新型教學平台，主要目的是希望能夠透過這種方式加強、加速學生學習程式的效率，並能夠快速實作課程中所教導的邏輯概念。本報告記錄暑期實習期間之工作事項，包含 Online Judge 的架設以及後續的修改或是遇到的問題，並對於實習成果進行簡單的闡述與心得。

目錄

| | |
|---|----|
| 致謝詞..... | 2 |
| 摘要 | 2 |
| 一、 如何在 Linux 環境下部署青島大學所開發之 Online Judge—QDUOJ。4 | |
| 快速了解 QDUOJ | 4 |
| Linux 環境下安裝流程 | 4 |
| 開始使用 | 5 |
| 二、啟動 OJ 後以 root 身分登入管理者後台介面，可修改內容、管理使用者、 題目等設定。 | 6 |
| 基本設定..... | 6 |
| 試題..... | 8 |
| 比賽..... | 11 |
| 三、修改前端頁面並更新到所發布的頁面。 | 14 |
| 連線到遠端伺服器 | 17 |
| 從本地傳送資料到遠端伺服器 | 17 |
| 五、修改埠號、架設網站並使其成功運行。 | 17 |
| 六、 申請 SSL 憑證將 http 網頁改為安全連線的 https 網頁。(未完成) ... | 19 |
| 方法 1：使用 ZeroSSL 平台申請核發 SSL 憑證並安裝 | 19 |
| 方法 2：使用 Let's Encrypt 和 certbot 套件核發憑證自簽並安裝（以 nginx 伺服器為例） | 22 |
| 七、Vim 簡易使用介紹 | 26 |
| 八、參考資料 | 27 |
| 實習成果及貢獻、結論、建議事項、學習心得 | 28 |

一、如何在 Linux 環境下部署青島大學所開發之 Online Judge—QDUOJ。

● 快速了解 QDUOJ

由青島大學所開發的開源 Online Judge 線上評測系統。管理員可建立題目、比賽提供學習使用。題目可自由新增、編輯，也可以選擇要使用什麼語言，語言支持 C, C++, Java, Python2, Python3。前端使用 Vue 框架，後端使用 Django，系統模組化。

參考 github 文件：<https://github.com/QingdaoU/OnlineJudge>

● Linux 環境下安裝流程

參考部署文件：<https://github.com/QingdaoU/OnlineJudgeDeploy>

1. 安裝需要的套件，開啟終端機輸入

```
sudo apt-get update && sudo apt-get install -y vim python3-pip curl git
```

```
pip3 install --upgrade pip
```

```
pip install docker-compose
```

2. 安裝 Docker

```
sudo apt-get install docker.io
```

安裝完成後查看是否正常啟動

```
service docker status
```

正常須為綠燈 active(running)

建立 docker 群組

```
sudo groupadd docker
```

將自己的使用者帳號加入 docker 群組（\$USER 請替換成自己的帳號）

```
sudo usermod -aG docker $USER 或是
```

```
sudo gpasswd -a $USER docker
```

更新 docker 群組資訊

```
newgrp docker
```

重新啟動 Docker 服務

```
sudo systemctl restart docker
```

查看 Docker 版本資訊

```
docker version
```

正常輸出須出現 Client: 以及 Server: 相關訊息

3. 開始安裝 QDUOJ

選擇記憶體充足的位置將 github 文件拉到本機

```
git clone -b 2.0 https://github.com/QingdaoU/OnlineJudgeDeploy.git  
&& cd OnlineJudgeDeploy
```

啟動服務

```
docker-compose up -d
```

等待安裝完畢並出現 4 個綠色 done，代表安裝完成，可以輸入

```
docker ps -a
```

查看容器狀態，若 STATUS 都正常代表已經啟動成功

● 開始使用

透過瀏覽器存取伺服器或本機的 http 80 port 或是 https 443 port 即可看到 OJ 頁面進行使用。

註冊頁面輸入帳號、密碼、email 以及驗證碼進行註冊（無經過信箱驗證即刻啟用）。

※注意：使用前請記得打開 docker 服務，不然可能會失敗

二、啟動 OJ 後以 root 身分登入管理者後台介面，可修改內容、管理使用者、題目等設定。

初始 root 帳號/密碼：root/rootroot

登入後請及時修改密碼

- 基本設定
- 使用者管理

| ID | Username | Create Time | Last Login | Real Name | Email | User Type | Option |
|----|------------|-------------------|--------------------|-----------|------------|--------------|-----------------|
| 5 | [REDACTED] | 2023-8-9 21:35:07 | 2023-8-9 21:35:18 | | [REDACTED] | Regular User | [Edit] [Delete] |
| 1 | root | 2023-8-4 12:14:03 | 2023-8-10 11:30:26 | | | Super Admin | [Edit] [Delete] |

可以使用 .csv 檔匯入複數使用者，也可以依起始號碼直接生成使用者在最上方可以看到已經存在的使用者，可以進行編輯或是刪除，按下編輯按鈕可以看到下方畫面，帳號類型分為下面 3 種：

Regular User、Admin、Super Admin。

使用者名稱: [REDACTED] 真實姓名: [REDACTED]

使用者 E-mail: [REDACTED] 使用者密碼: [REDACTED]

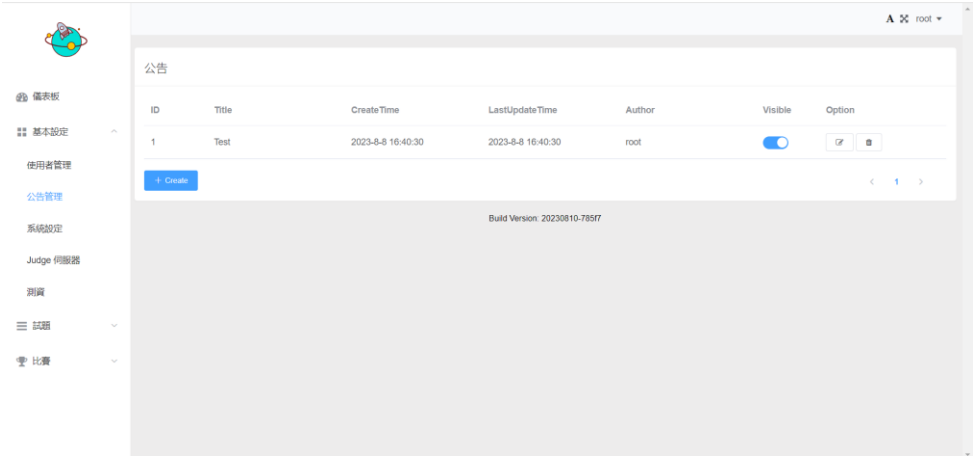
帳號類型: Regular User (dropdown menu open showing Regular User, Admin, Super Admin)

試題權限: None (dropdown menu)

兩步驗證: [ON] 是否禁用: [OFF]

[取消] [存檔]

公告管理



可以按下藍色的 **Create** 按鈕新增公告，並且可以設定是否可見。存檔後可以在列表中進行編輯或刪除。













Create Announcement

×

• 標題

標題

• 內容

H B I U A A   |        |   

|

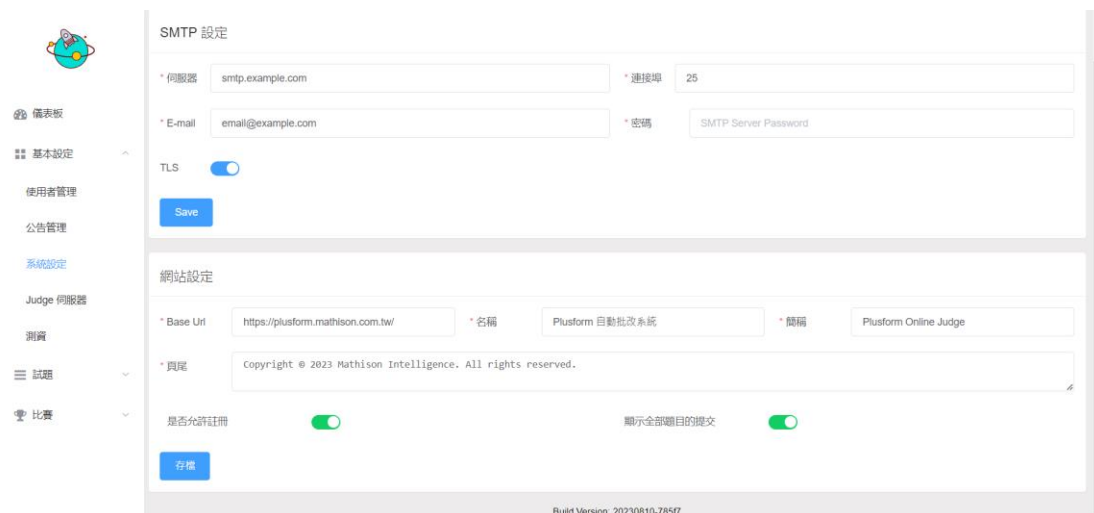
M1

是否可見 ☒

取消

存檔

系統設定



SMTP 設定

* 伺服器 * 連接埠

* E-mail * 密碼

TLS ☒

網站設定

* Base Url * 名稱 * 簡稱

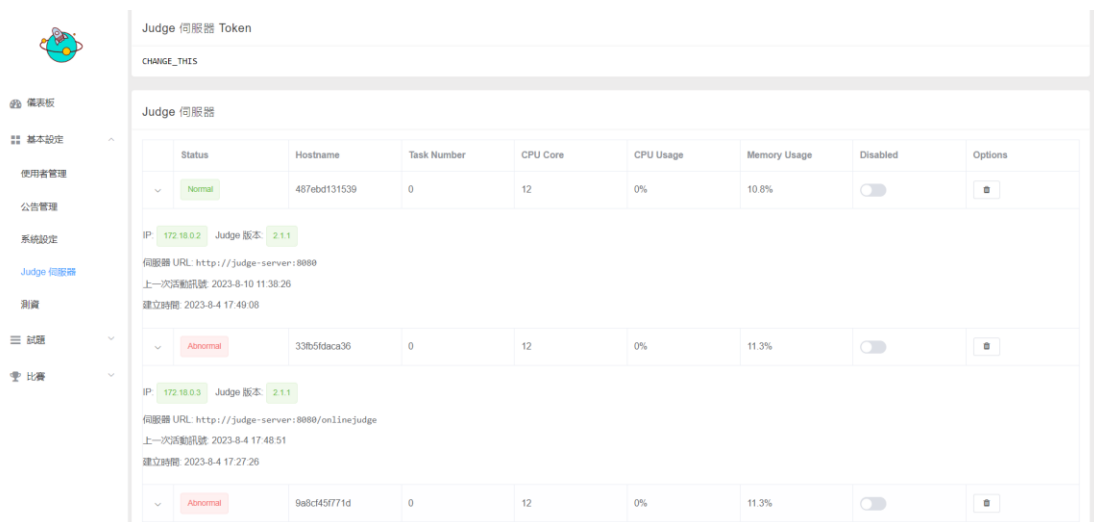
* 頁尾

是否允許註冊 ☒ 顯示全部題目的提交 ☒

Build Version: 20230810-785f7

系統設定頁面中，可自行更改網站的 Base Url、名稱、簡稱以及頁尾。

Judge 伺服器



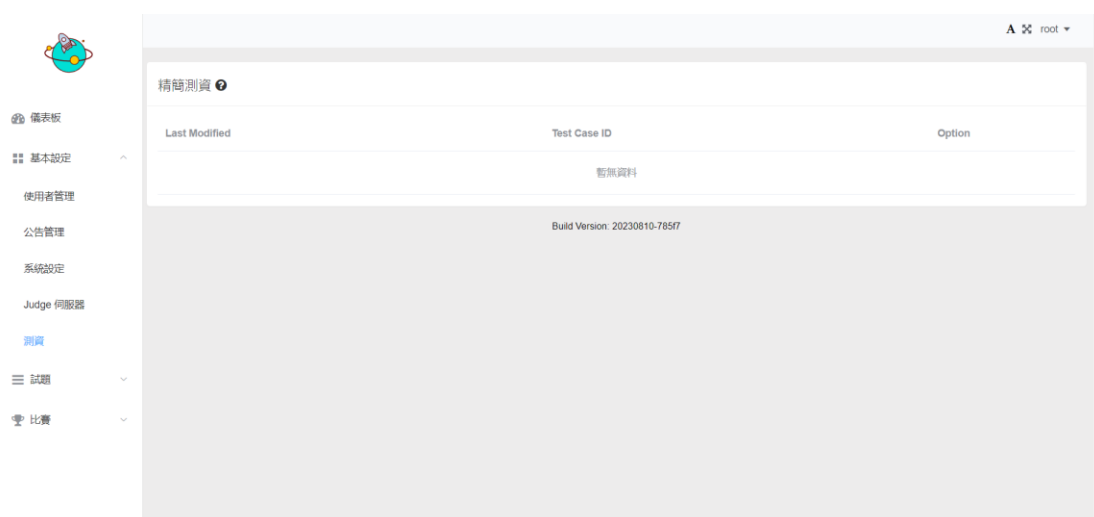
Judge 伺服器 Token

CHANGE_THIS

Judge 伺服器

| Status | Hostname | Task Number | CPU Core | CPU Usage | Memory Usage | Disabled | Options |
|--|--------------|-------------|----------|-----------|--------------|--------------------------|----------------------------------|
| Normal | 487ebd131539 | 0 | 12 | 0% | 10.8% | <input type="checkbox"/> | <input type="button" value="ⓘ"/> |
| IP: 172.18.0.2 Judge 版本: 2.1.1 伺服器 URL: http://judge-server:8080 上一次活動訊號: 2023-8-10 11:38:26 建立時間: 2023-8-4 17:49:08 | | | | | | | |
| Abnormal | 33b5fdca36 | 0 | 12 | 0% | 11.3% | <input type="checkbox"/> | <input type="button" value="ⓘ"/> |
| IP: 172.18.0.3 Judge 版本: 2.1.1 伺服器 URL: http://judge-server:8080/onlinejudge 上一次活動訊號: 2023-8-4 17:48:51 建立時間: 2023-8-4 17:27:26 | | | | | | | |
| Abnormal | 9a8c45f771d | 0 | 12 | 0% | 11.3% | <input type="checkbox"/> | <input type="button" value="ⓘ"/> |

測資



測資

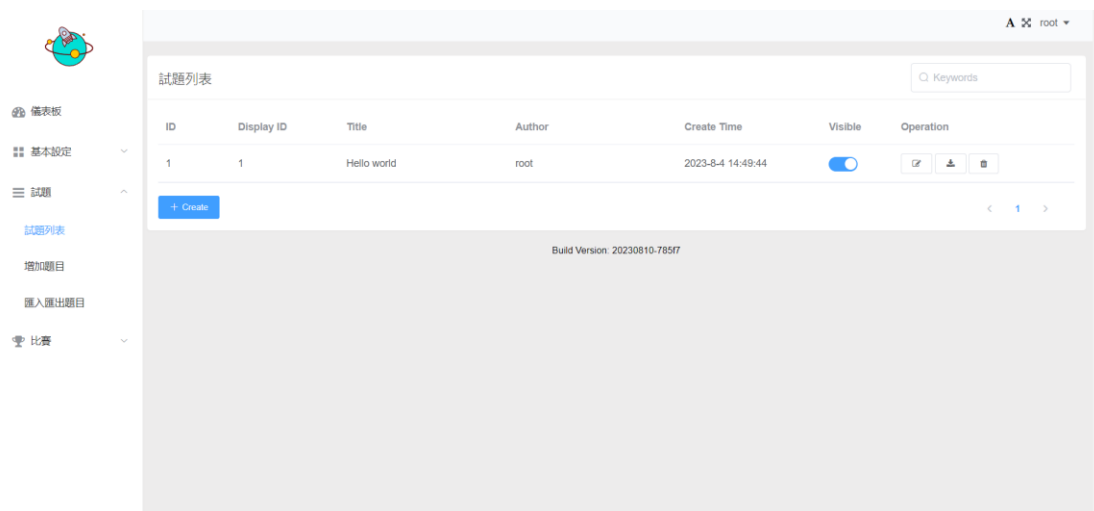
精簡測資 ⓘ

| Last Modified | Test Case ID | Option |
|---------------|--------------|--------|
| 暫無資料 | | |

Build Version: 20230810-785f7

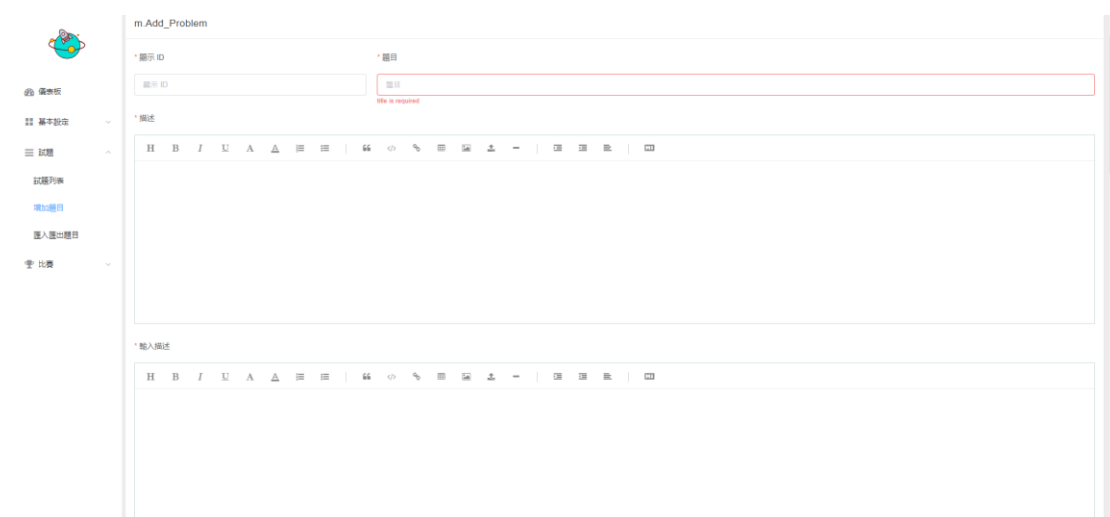
● 試題

試題列表



按下藍色 Create 按鈕會跑出與增加題目一樣的頁面

增加題目



The screenshot shows the 'Output Description' section of a problem configuration interface. It includes a text editor for the problem description, a 'Time Limit (ms)' field set to 1000, a 'Memory Limit (MB)' field set to 256, and a 'Difficulty' dropdown set to '初級'. There are also checkboxes for '是否可見' (Visible) and '分享提交' (Share Submission), and a 'Tag' field with a dropdown menu. Below these are checkboxes for '可選程式語言' (Selectable Languages) including C, C++, Java, Python2, Python3, Go, and JavaScript. A 'Sample' section is visible at the bottom.

The 'Code Template' section shows a C++ code template with the following content:

```

1 //PREPEND BEGIN
2 #include <iostream>
3 //PREPEND END
4
5 //TEMPLATE BEGIN
6 int add(int a, int b) {
7     // Please fill this blank
8     return _____;
9 }
10 //TEMPLATE END
11
12 //APPEND BEGIN
13 int main() {
14     std::cout << add(1, 2);
15     return 0;
16 }
17 //APPEND END

```

Below the code template, there are checkboxes for 'JavaScript', 'Python3', and 'Special Judge'. There is also a checkbox for '使用 Special Judge' (Use Special Judge).

可依需求設定題目 ID、題目名稱、題目描述、輸入輸出描述、時間及記憶體限制、難度等級、自訂標籤、所使用的語言、輸入輸出範例、加入程式語言模板等。

The screenshot shows the 'Test Type' and 'IO Type' sections of a problem configuration interface. The 'Test Type' section has radio buttons for 'ACM' (selected) and 'OI', and a 'Choose File' button. The 'IO Type' section has radio buttons for 'Standard IO' (selected) and 'File IO'. Below these are fields for '輸入' (Input), '輸出' (Output), and '分數' (Score). There is also a '來源' (Source) field and a '存檔' (Save) button.

The 'Test Type' section also includes a 'Special Judge' checkbox and a '使用 Special Judge' checkbox.

The 'IO Type' section includes a 'Standard IO' radio button and a 'File IO' radio button.

The 'Input' field is labeled '輸入' and the 'Output' field is labeled '輸出'. The 'Score' field is labeled '分數'.

The 'Source' field is labeled '來源'.

The 'Save' button is labeled '存檔'.

The 'Build Version' is 20230810-7857.

測試類型可選擇

ACM：全對才給分。

OI：對測資就給分，有部分給分。

加入評分用的測資，按下 Choose File 上傳壓縮檔，須注意壓縮檔內檔案的擺放方式以及檔案命名方式：

```
→ testcase pwd
/tmp/testcase
→ testcase tree
.
├── 1.in
├── 1.out

0 directories, 2 files
```

檔案名稱為：編號.in、編號.out

有多組測試範例可以包成 1.in、1.out、2.in、2.out…等，壓縮在同一壓縮檔內。

測試類型

測資

IO 類型

☒ ACM ☐ OI

Choose File

☒ Standard IO ☐ File IO

輸入

輸出

分數

1.in


1.out

100

來源

來源

匯入匯出題目



備表板

基本設定

試題

試題列表

增加題目

匯入匯出題目

比賽

Export Problems (beta)

Q Keywords

| ID | DisplayID | Title | Author | Create Time |
|---------------------------------------|-----------|-------------|--------|-------------------|
| <input checked="" type="checkbox"/> 1 | 1 | Hello world | root | 2023-8-4 14:49:44 |

Export

Import QDUOJ Problems (beta)

Choose File

Upload

Import FPS Problems (beta)

Choose File

Upload

Build Version: 20230810-785f7

● 比賽

比賽列表

Contest Problem List

| ID | Display ID | Title | Author |
|--|------------|-------|--------|
| 暫無資料 | | | |
| <div><div>+ Create</div><div>+ Add From Public Problem</div></div> | | | |

比賽建立後，可以進入編輯所使用的題目。

| Add Contest Problem | | | | × |
|-----------------------|-----------|-------------|--------------|---|
| <div>Q Keywords</div> | | | | |
| ID | DisplayID | Title | option | |
| 2 | 2 | Test | <div>+</div> | |
| 1 | 1 | Hello world | <div>+</div> | |

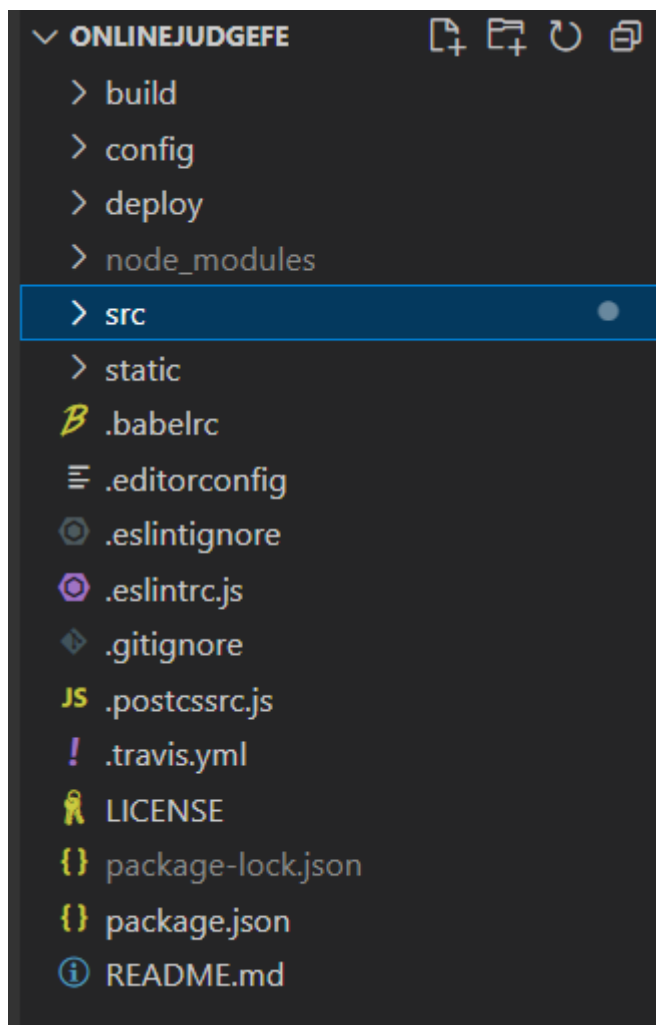
可以直接從題目列表加入已有的題目，或是再新增、刪除題目。

三、修改前端頁面並更新到所發布的頁面。

- QDUOJ 前後端分離，可 fork 一份 QDUOJ 的前端頁面到自己的 github 方便進行版本管理。

Github 文件：<https://github.com/QingdaoU/OnlineJudgeFE>

- 將檔案一樣拉到本機端，可以使用自己習慣的 IDE 編輯
git clone <https://github.com/QingdaoU/OnlineJudgeFE>.git
- 檔案層級說明



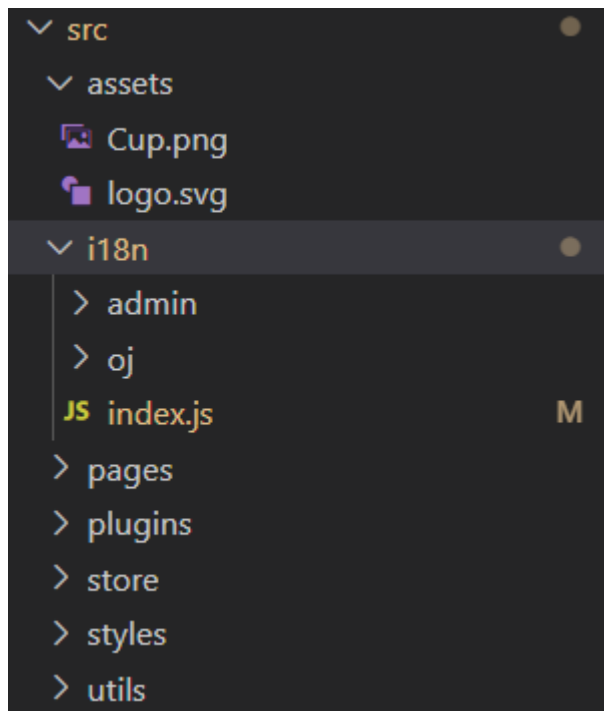
build：用於 webpack，將前端打包用

config：模組的一些設定，run dev 所使用的埠號等

deploy：放 docker 和 nginx 的設定檔等

node_modules：npm 需要的一些模組

src：前端各個頁面和各種可見的資料，都在這個資料夾更改前端



assets：放置一些圖片，像 logo 等

```
src > i18n > admin > JS zh-TW.js > [e] m
1  export const m = {
2    // SideMenu.vue
3    Dashboard: '儀表板',
4    General: '基本設定',
5    User: '使用者管理',
6    Announcement: '公告管理',
7    System_Config: '系統設定',
8    Judge_Server: 'Judge 伺服器',
9    Prune_Test_Case: '測資',
10   Problem: '試題',
11   FromFile: '讀取檔案',
```

i18n：存放語言對應的翻譯，預設有英文、簡體中文和繁體中文，可以在 index.js 修改想使用的語言

pages：放置各頁面的.Vue 檔模組，包含頁面的按鍵或是 header 等，也存放 App.Vue 和 router 等檔案。

- 修改後的前端如何更新到網頁

1. 在本機瀏覽器確認運行狀況
(在 OnlineJudgeFE 目錄下)

```
export TARGET=http://Your-backend
```

```
npm run dev
```

2. 確認後將新版本進行打包

```
npm run build
```

build 完畢後會產生 dist 資料夾

3. 將 dist 資料夾複製到 OnlineJudgeDeploy/data/backend 目錄下

4. 修改 docker-compose.yml 檔案

```
cd 路徑/OnlineJudgeDeploy
```

```
sudo vim docker-compose.yml
```

5. 在 oj-backend 下的 volumes 欄位，新增使用的 dist 路徑（如圖）

```
volumes:
  - ./data/backend:/data
  - /home/changmc/new/OnlineJudgeDeploy/data/backend/dist:/app/dist
environment:
```

※注意：路徑不用加前面的「.」

6. 修改完畢後，確認在 OnlineJudgeDeploy 目錄下，重新啟動服務

```
docker-compose up -d
```

更新後可在瀏覽器確認更改後的網頁

四、使用 ssh 連線登入伺服器、使用 ssh 連線在本地與遠端傳遞檔案。

- 連線到遠端伺服器

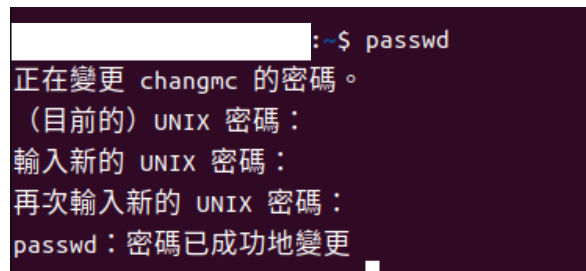
ssh 帳號@主機

例如：

ssh changmc@192.xxx.xxx.xxx -p 2222

-p 參數代表使用 2222 埠號連線進入

登入後輸入 passwd 更改密碼



```
changmc@192.xxx.xxx.xxx:~$ passwd
正在變更 changmc 的密碼。
(目前的) UNIX 密碼:
輸入新的 UNIX 密碼:
再次輸入新的 UNIX 密碼:
passwd: 密碼已成功地變更
```

- 從本地傳送資料到遠端伺服器

1. 安裝 ssh 所需套件

[本地端] sudo apt-get install openssh-client

[遠端] sudo apt-get install openssh-server

2. 傳送指令

[本地端] scp FileName user@remote:/home/new/

說明：

FileName: 預複製的檔案名稱。

user: 遠端使用者。

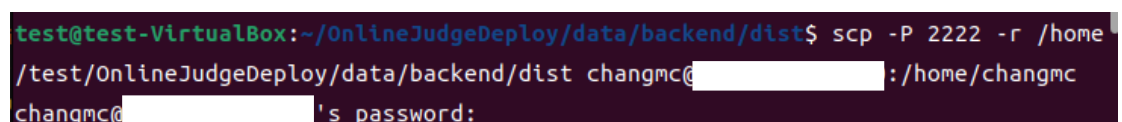
remote: 遠端伺服器的位址，可以是 IP 也可以是網域名稱(Domain name)。

/home/new/: 遠端伺服器內的檔案位置。

例子：

scp test.txt changmc@192.xxx.xxx.xxx:/home/changmc/new

使用-P 參數可設定連線埠號 一次傳送多個檔案可加-r 參數



```
test@test-VirtualBox:~/OnlineJudgeDeploy/data/backend/dist$ scp -P 2222 -r /home
/test/OnlineJudgeDeploy/data/backend/dist changmc@192.xxx.xxx.xxx:/home/changmc
changmc@192.xxx.xxx.xxx's password:
```

五、修改埠號、架設網站並使其成功運行。

OJ 預設放在 80 或 443 下，可能會與原伺服器有埠號衝突的問題。因此

需修改 OJ 的設定。

下圖：0.0.0.80: bind: address already in use

```
ERROR: for oj-backend Cannot start service oj-backend: driver failed programmin
g external connectivity on endpoint oj-backend (a0d3f17a8dc5da75a3626ce4d71c6457
d52c526f9a31353d75d7c8a78fdce05e): Error starting userland proxy: listen tcp4 0.
0.0.0:80: bind: address already in use

ERROR: for oj-backend Cannot start service oj-backend: driver failed programmin
g external connectivity on endpoint oj-backend (a0d3f17a8dc5da75a3626ce4d71c6457
d52c526f9a31353d75d7c8a78fdce05e): Error starting userland proxy: listen tcp4 0.
0.0.0:80: bind: address already in use
ERROR: Encountered errors while bringing up the project.
```

在 OnlineJudgeDeploy 目錄下，編輯 docker-compose.yml

```
sudo vim docker-compose.yml
```

在 oj-backend 下的 ports:

修改要使用的埠號（圖中修改成 8080）0.0.0.0:本機埠號:容器埠號

```
- JUDGE_SERVER_TOKEN=CHANGE_THIS
# - FORCE_HTTPS=1
# - STATIC_CDN_HOST=cdn.oj.com
ports:
  - "0.0.0.0:8080:8000"
  - "0.0.0.0:443:1443"
```

修改完畢後記得更新設定

```
docker-compose up -d
```

這時再打開伺服器的 8080 port，可以看到網站已成功運行。

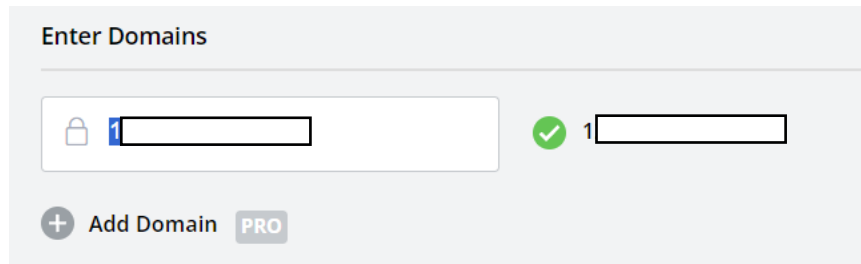
六、申請 SSL 憑證將 http 網頁改為安全連線的 https 網頁。(未完成)

- 為什麼需要 SSL 憑證

SSL 憑證可以加強網站連線的安全性，有 SSL 憑證的網站會是 https 開頭，如果網站為 http 開頭，可能會有瀏覽不安全網頁的風險。

- 方法 1：使用 ZeroSSL 平台申請核發 SSL 憑證並安裝

1. 於 ZeroSSL 註冊帳號
2. 輸入合格的 domain name 或是 ip 位置

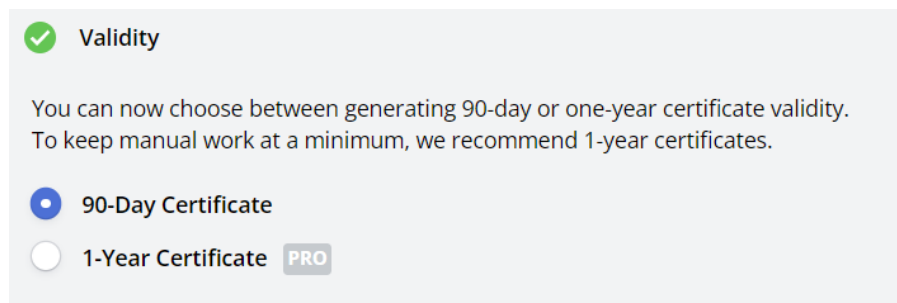


Enter Domains

+ Add Domain PRO

3. 選擇 90 天方案（免費）

4. 不更動

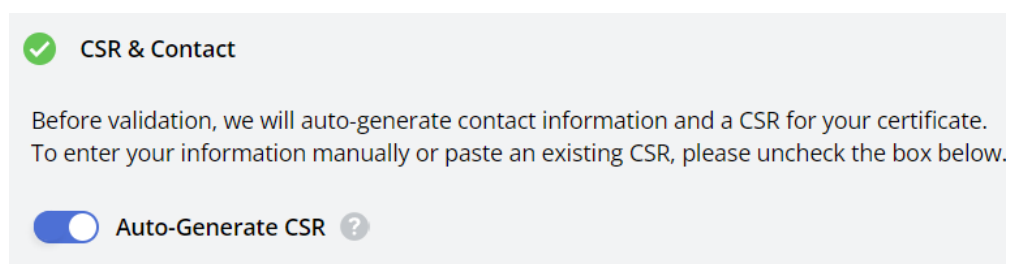


✓ Validity

You can now choose between generating 90-day or one-year certificate validity. To keep manual work at a minimum, we recommend 1-year certificates.

☒ 90-Day Certificate

☐ 1-Year Certificate PRO



✓ CSR & Contact

Before validation, we will auto-generate contact information and a CSR for your certificate. To enter your information manually or paste an existing CSR, please uncheck the box below.

☒ Auto-Generate CSR ?

5. 預設選擇 Free，根據要求選擇其他方案

Finalize Your Order

Based on your selection of a 90-Day SSL Certificate you are fine staying on the Free Plan. To create and validate your SSL Certificate, please click "Next Step" below.

| Free | Basic | Premium | Business |
|--|--|---|---|
| \$0 / month | \$10 / month or \$8 if billed yearly | \$50 / month or \$40 if billed yearly | \$100 / month or \$80 if billed yearly |
| Selected | Select | Select | Select |
| <ul style="list-style-type: none"> 3 90-Day Certificates 1-Year Certificates Multi-Domain Certs 90-Day Wildcards 1-Year Wildcards REST API Access Technical Support | <ul style="list-style-type: none"> 3 90-Day Certificates 1-Year Certificates Multi-Domain Certs 90-Day Wildcards 1-Year Wildcards REST API Access Technical Support | <ul style="list-style-type: none"> 10 90-Day Certificates 1-Year Certificates Multi-Domain Certs 90-Day Wildcards 1-Year Wildcards REST API Access Technical Support | <ul style="list-style-type: none"> 25 90-Day Certificates 1-Year Certificates Multi-Domain Certs 90-Day Wildcards 1-Year Wildcards REST API Access Technical Support |

Next Step →

6. 驗證方法選擇 HTTP File Upload

HTTP File Upload

Follow the steps below

To verify your domain using HTTP File Upload, please follow the steps below:

- Download your Auth File using the following link: [Download Auth File](#)
- Upload the Auth File to your HTTP server under: `/.well-known/pki-validation/`
- Make sure your file is available under the following link: <http://validation/5D5DD2B18590BC9CA3C27B5AE8D2A0C0.txt>
- Click "Next Step" to continue.

7. 下載 Auth File 並按指示建立資料夾

```
changmc@lzhong-at-fin:~/new/OnlineJudgeDeploy/data/backend/ssl/.well-known$ ls -la
.  ..  pki-validation
changmc@lzhong-at-fin:~/new/OnlineJudgeDeploy/data/backend/ssl/.well-known$
```

8. 將準備好的 txt 文件 copy 至目錄

```

changmc@lzhong-ai-fin:~$ sudo cp *txt new/OnlineJudgeDeploy/data/backend/ssl/.well-known
/pki-validation
changmc@lzhong-ai-fin:~$ cd new/OnlineJudgeDeploy/data/backend/ssl/.well-known/pki-validation
changmc@lzhong-ai-fin:~/new/OnlineJudgeDeploy/data/backend/ssl/.well-known/pki-validation$ ls -a
.  ..  5D5DD2B18590BC9CA3C27B5AE8D2A0C0.txt

```

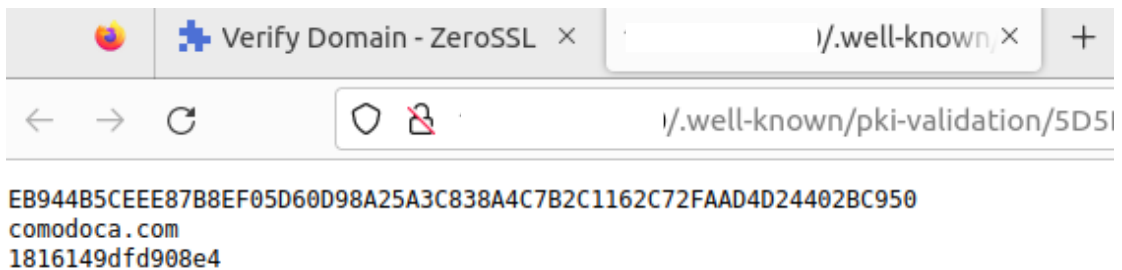
網址需要 ip+:8080 才能看到文件資料，驗證失敗(網頁顯示 404)

9. 改放在 port80 的目錄下

```

changmc@lzhong-ai-fin:~$ sudo cp -r .well-known /var/www/html
changmc@lzhong-ai-fin:~$ cd /var/www/html
changmc@lzhong-ai-fin:/var/www/html$ ls -a
.  ..  index.html  index.nginx-debian.html  .well-known
changmc@lzhong-ai-fin:/var/www/html$

```

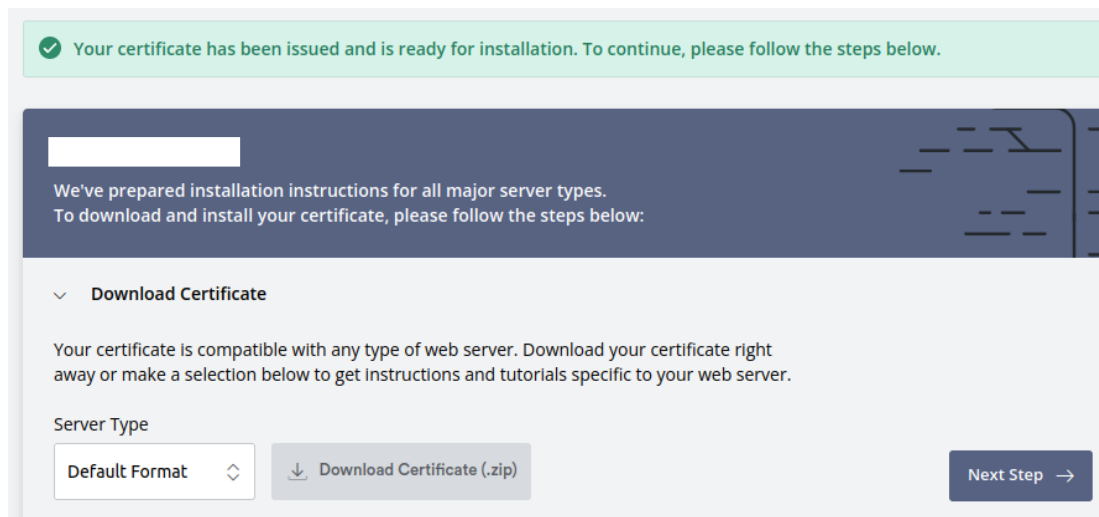


10. 驗證成功，下載壓縮檔會得到 3 個檔案，分別是

certificate.crt

ca_bundle.crt

private.key



11. 將準備好的檔案放到指定資料夾，依據 OS 參考 ZeroSSL 的說明

<https://help.zerossll.com/hc/en-us/articles/360060120373-Installing-SSL-Certificate-on-Ubuntu>

```
changmc@lzrong-ai-fin:~$ sudo cp *.crt /etc/ssl
changmc@lzrong-ai-fin:~$ cd /etc/ssl
changmc@lzrong-ai-fin:/etc/ssl$ ls
ca_bundle.crt  certificate.crt  certs  openssl.cnf  private
```

12. 取消 docker-compose.yml 中 FORCE_HTTPS=1 的註解

```
- FORCE_HTTPS=1
# - STATIC CDN HOST=cdn.oj.com
```

- 遇到的錯誤：

1. 因存取權限無法將 private.key 放入 private 資料夾
2. 無法重新啟動 apache2 或是 nginx 服務

- 方法 2：使用 Let's Encrypt 和 certbot 套件核發憑證自簽並安裝（以 nginx 伺服器為例）

1. 下載和安裝 certbot 套件

```
sudo apt-get update -y
```

```
sudo apt-get install certbot
```

```
sudo apt-get install certbot python3-certbot-nginx -y
```

2. 需要設定 DNS 的 A Record（*需有 DNS 所有權才行）和修改 conf 檔

注意路徑不一定相同，請按照自己主機的路徑輸入指令

```
sudo vim /etc/nginx/sites-available/nginx.conf
```

修改如下

```
server {
    listen 80;
    listen [::]:80;
    ##Certbot會看的設定如下『server_name』
    server_name imkjie.com www.imkjie.com;
    root /var/www/kjnotes;

    # ...
}
```

修改後輸入

```
sudo nginx -t
```

檢查執行有沒有錯誤並重新載入 conf 檔

看到以下訊息代表設定沒有出錯：

```
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
```

```
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

重新載入 nginx 服務

```
sudo systemctl reload nginx
```

3. 使用 domain name 申請 SSL 憑證

以下 email@example 以及 example.com 請替換成自己的

```
sudo certbot certonly --nginx --email email@example.com --agree-tos
-d example.com -d www.example.com
```

申請過程可能會詢問一些問題，比如會詢問是否需要收到 Let's Encrypt 組織相關的郵件，選擇 y 或 n

4. 憑證申請成功後，會出現『Congratulations!』的訊息，而申請成功的憑證會存放在『/etc/letsencrypt/live/example.com』目錄中。

IMPORTANT NOTES:

- Congratulations! Your certificate and chain have been saved at:
/etc/letsencrypt/live/plusform.mathison.com.tw/fullchain.pem
Your key file has been saved at:
/etc/letsencrypt/live/plusform.mathison.com.tw/privkey.pem
Your cert will expire on 2023-11-08. To obtain a new or tweaked version of this certificate in the future, simply run certbot again. To non-interactively renew *all* of your certificates, run "certbot renew"
- If you like Certbot, please consider supporting our work by:

Donating to ISRG / Let's Encrypt: <https://letsencrypt.org/donate>
Donating to EFF: <https://eff.org/donate-le>

目錄下應有 4 個檔案，分別是：

cert.pem：伺服器端的憑證（終端憑證）；透過 Let's Encrypt 中續憑證產生和簽發給你的公鑰憑證

chain.pem：Let's Encrypt 的中續憑證

fullchain.pem：此憑證包含了 cert.pem 和 chain.pem 兩個檔案的內容

privkey.pem：憑證的私鑰

5. 設定網路伺服器的 conf 設定

```
sudo vim /etc/nginx/sites-available/nginx.conf
```

```
server {  
    listen 80;  
    listen [::]:80;  
    server_name example.com www.example.com;  
    return 301 https://example.com$request_uri;  
}
```

請自行將 root /var/www/example 替換至 Nginx 存放網頁的根目錄，
及 example.com 替換成自己的域名


```
#HTTPS-有www轉址到沒有www
server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    server_name www.example.com;
    ssl_certificate /etc/letsencrypt/live/example.com/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/example.com/privkey.pem;
    ssl_ecdh_curve X25519:secp384r1;
    ssl_session_cache shared:SSL:50m;
    ssl_session_timeout 1440m;
    ssl_session_tickets off;
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers TLS13-AES-256-GCM-SHA384:TLS13-CHACHA20-POLY1305-SHA256;
    ssl_prefer_server_ciphers on;
    ssl_stapling on;
    ssl_stapling_verify on;
    ssl_trusted_certificate /etc/letsencrypt/live/example.com/chain.pem;
    add_header Strict-Transport-Security "max-age=31536000; preload";
    return 301 https://example.com$request_uri;
}
```

```
#HTTPS-沒有www-最終網址
server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    server_name example.com;
    root /var/www/html;
    ssl_certificate /etc/letsencrypt/live/example.com/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/example.com/privkey.pem;
    ssl_ecdh_curve X25519:secp384r1;
    ssl_session_cache shared:SSL:50m;
    ssl_session_timeout 1440m;
    ssl_session_tickets off;
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers TLS13-AES-256-GCM-SHA384:TLS13-CHACHA20-POLY1305-SHA256;
    ssl_prefer_server_ciphers on;
    ssl_stapling on;
    ssl_stapling_verify on;
    ssl_trusted_certificate /etc/letsencrypt/live/example.com/chain.pem;
    add_header Strict-Transport-Security "max-age=31536000; preload";

    # ...
}
```

6. 測試設定與執行有沒有錯誤

```
sudo nginx -t
```

重新載入 nginx 的 conf 檔

```
sudo systemctl reload nginx
```

- 遇到的錯誤：

Nginx 服務沒辦法重新啟動

沒有出現 https，SSL 申請失敗

七、Vim 簡易使用介紹

對任何想編輯的文件，輸入指令

`sudo vim 文件名稱`

- 按下 i、o、r 任一符號：進入插入模式才可編輯
- 按下 r：取代模式

才可以編輯文件內容，編輯完成後輸入 `esc` 鍵可以退出編輯模式

- 按下 /、:、? 任一符號：進入命令模式
- 使用 /+想搜尋的關鍵字，可以在檔案內搜尋關鍵字出現的位置

常用指令：

| 指令 | 功能 |
|-------------------|------------------|
| <code>:w</code> | 儲存 |
| <code>!:w</code> | 強制儲存 |
| <code>:q</code> | 離開 |
| <code>!:q</code> | 強制離開（如未儲存則不儲存修改） |
| <code>:wq</code> | 儲存並離開 |
| <code>!:wq</code> | 強制儲存並離開 |

八、參考資料

- huli (2020 年 03 月 20 日)。自己架一個 Online Judge 系統
(<https://blog.techbridge.cc/2020/03/20/build-your-own-online-judge/>)。
- OnlineJudge 2.0 (https://opensource.qduoj.com/#/judger/how_it_works)。
- yangj26952 (2017 年 12 月 09 日)。Day6：把 Docker Image Push 到 Docker Hub
(<https://ithelp.ithome.com.tw/articles/10191139>)。
- Vue.js 目錄結構 (<https://www.runoob.com/vue2/vue-directory-structure.html>)。
- ZeroSSL 教學，免費 SSL 憑證-申請與安裝步驟
(<https://kamadiam.com/zeross-1-tutorial/>)。
- [HTTPS] Ubuntu +SSL +正式的 CA 設定
(<https://n.sfs.tw/content/index/10939>)。
- KJie (2022 年 02 月 02 日)。Debian/Ubuntu 環境申請和套用 Let's Encrypt
免費 SSL 憑證教學 (<https://www.kjnotes.com/devtools/62>)。
- finch_xu (2018 年 08 月 18 日)。qduoj~前端~二次开发
(<https://www.cnblogs.com/finch-xu/archive/2018/08/18/9497735.html>)。
- 江大衛 (2021 年 03 月 29 日)。快速建立屬於你的 online judge
(<https://theriseofdavid.github.io/2021/03/29/linux/linux-build-online-judge/>)。
- 腳印網頁資訊設計 (2018 年 07 月 27 日)。vi、Vim 文字編輯器教學和常用
按鍵與指令表 (<https://footmark.com.tw/news/linux/centos/vi-vim/>)。

● 實習成果及貢獻

本次實習成果完成一個教學平台網站的雛形，課程網站可包含每個章節的教學影片，並依對應的課程內容在線上自動批改系統建立相關概念的題目，學生可以即時提交並檢視學習成果，課程概念結合題目實作可以更有效吸收各章節所教導之內容。

● 結論

搭建完 Online Judge 並使其成功運行，可透過模組化的文件進行前端的修改且更新網站頁面。顧慮到安全性瀏覽的部分並申請 SSL 憑證，但是在安裝上遇到問題所以這部分還尚未完成。與教學網站還沒互相連接，但可透過網址前往對應網頁。

● 建議事項

事前若能夠先了解 Docker、架設網站的基本觀念會更好，減少在相同的問題裡打轉的時間，也能增進效率。

● 學習心得

在本次實習過程中接觸到了許多在大學課程中沒有使用過的工具，也在開發過程中遇到了各種錯誤及問題，常常都在找尋解決問題的方法並耗費了許多時間。透過這次的經驗，深知自己的不足以及需要增進的能力，並感謝指導老師們提供的機會以及提醒，不只是技術層面，也學習到工作方面待人處事的道理。