



ghost removal的線上版



# 系統目標

- 透過網頁的UI介面，讓Ghost Removal的使用者有更便捷的操作過程。

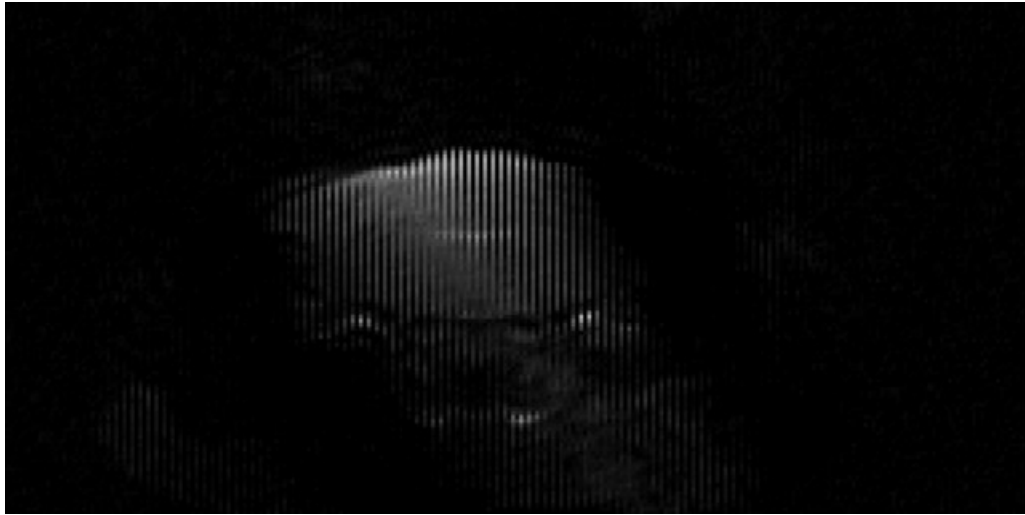


圖. Complex影像

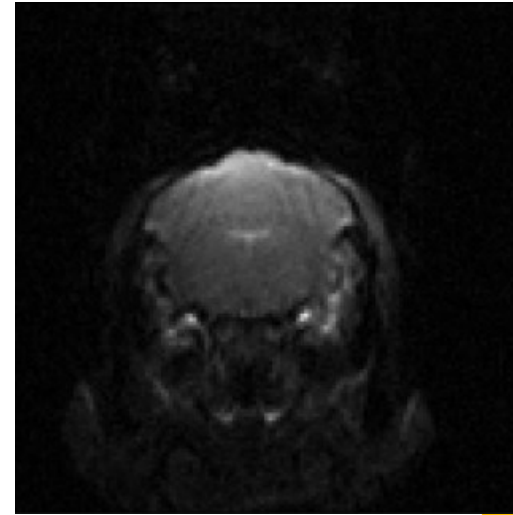


圖. 經過Ghost Removal的ground truth影像

# 系統架構

---

我們利用前端HTML和後端Flask進行網頁開發，並使用MySQL紀錄使用者資訊。



Python輕量級web應用框架



為網頁內容的一種標記語言



關聯式資料庫管理系統

# 系統架構—檔案結構

系統的檔案結構如右。

templates：存放網頁的html檔案

static：存放所有靜態的檔案，例如js、css

app.py：主要運行Flask的後端程式，連結網頁與資料庫

sql.py：server在固定時間會執行此Python檔案，刪除資料庫中超過期限的data

```
├── templates
│   └── 所有的網頁html檔案
├── static
│   ├── css
│   │   └── 所有的網頁css檔案
│   └── nifti-reader.js
├── app.py
└── sql.py
```

圖. 檔案結構

# Flask後端—form

將html的form利用action傳送到相對應的路徑的函式去執行。

若同一個html當中有不同的form，在函式執行時會用表單中的內容去判斷應該要執行的動作。

```
@app.route('/', methods=['GET', 'POST'], strict_slashes=False)
def home():
    global mem
    > if request.method == "POST": ...

    if mem == 0:
        return render_
    else:
        return render_

# 若是用NIFTI的檔案上傳
if "NIFTI" in request.files:
    # 使用多線程
    thr = Thread(target=dobNIFTI, args=[mem, request.files["NIFTI"].filename, file_dir, request.files["mask"].filename, request.
form["seg"], request.form["cost"], request.form["method"], request.files["bvec"].filename])
    thr.start()
    return redirect(url_for('membersonly'))

# 若是用bruker的方式上傳檔案
elif "bruker" in request.files:
    thr = Thread(target=dobBruker, args=[mem, request.files["bruker"].filename, file_dir, request.files["reco"].filename,
request.files["method_file"].filename, request.files["mask"].filename, request.form["seg"], request.form["cost"], request.
form["method"], request.files["bvec"]])
    thr.start()
    return redirect(url_for('membersonly'))
```

圖. 範例程式碼-判斷檔案內容

# 自動化系統

利用crontab排程與設定terminal，讓作業系統在開機時可以自動架設網站。

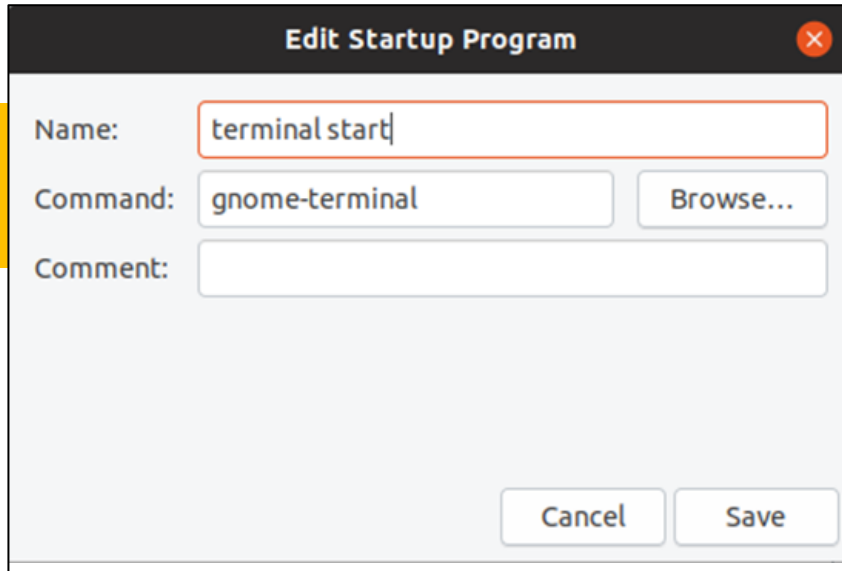


圖. 開機自動開啟terminal

```
ahdir= apsearchn -atni_help_dir
if [ -f "$ahdir/all_progs.COMP.bash" ]
then
. $ahdir/all_progs.COMP.bash
fi

export PATH=$PATH:/home/kazamiasurada/abin
export R_LIBS=$HOME/R
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:~/Qt5.12.12/5.12.12/gcc_64/lib

/usr/local/bin/python3.10 ~/Desktop/mri/app.py
```

圖. 開啟terminal自動執行app.py

```
kazamiasurada@kazmiasurada-VirtualBox: ~
@reboot /etc/init.d/cron start;
10 10 * * * find ~/Desktop/mri -type d -name "upload*" -atime +7 -exec rm -rf {} \;
10 10 * * * /usr/local/bin/python3.10 ~/Desktop/mri/sql.py
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
```

圖. crontab排程(自動刪除已經過期的檔案)

# 成果

註冊：為統計使用者，紀錄其姓名、信箱及所屬單位。

登入：利用信箱作為帳號登入(不需要密碼)。

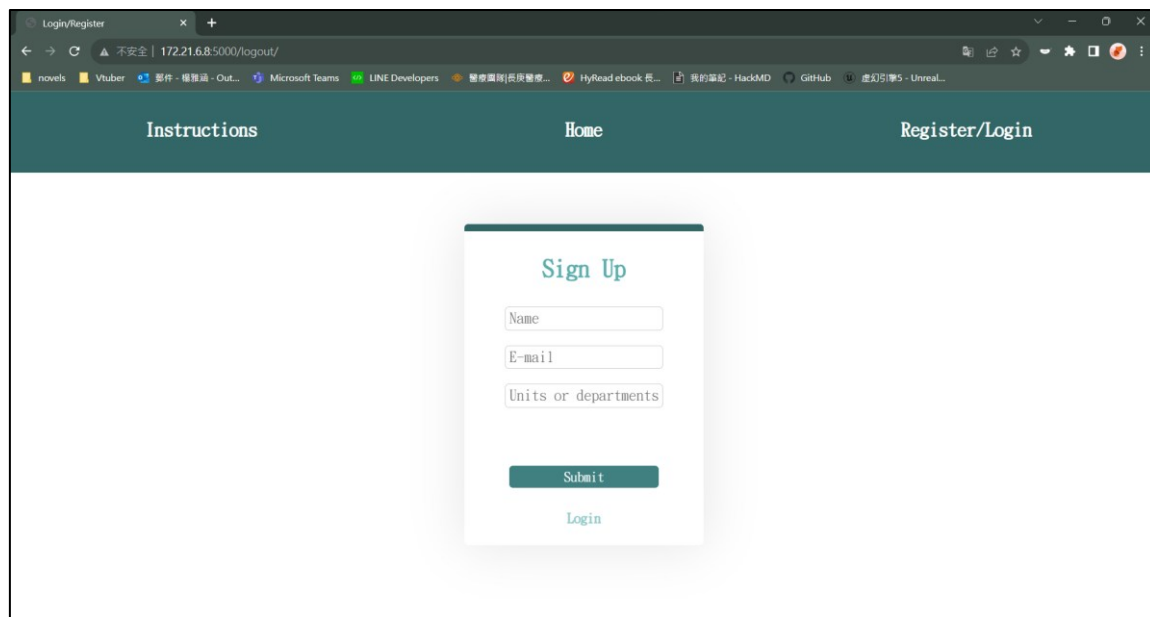


圖. 註冊

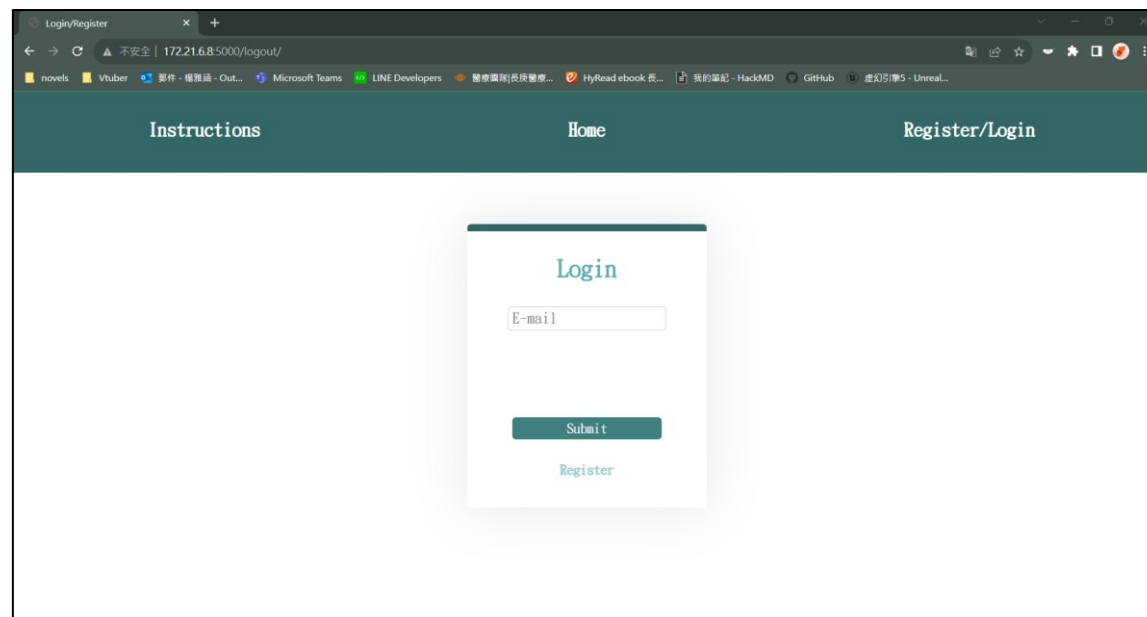


圖. 登入

# 成果

輸入：1. NIFTI file  
2. bruker data

輸出：輸出15\_gr.nii供使用者下載。

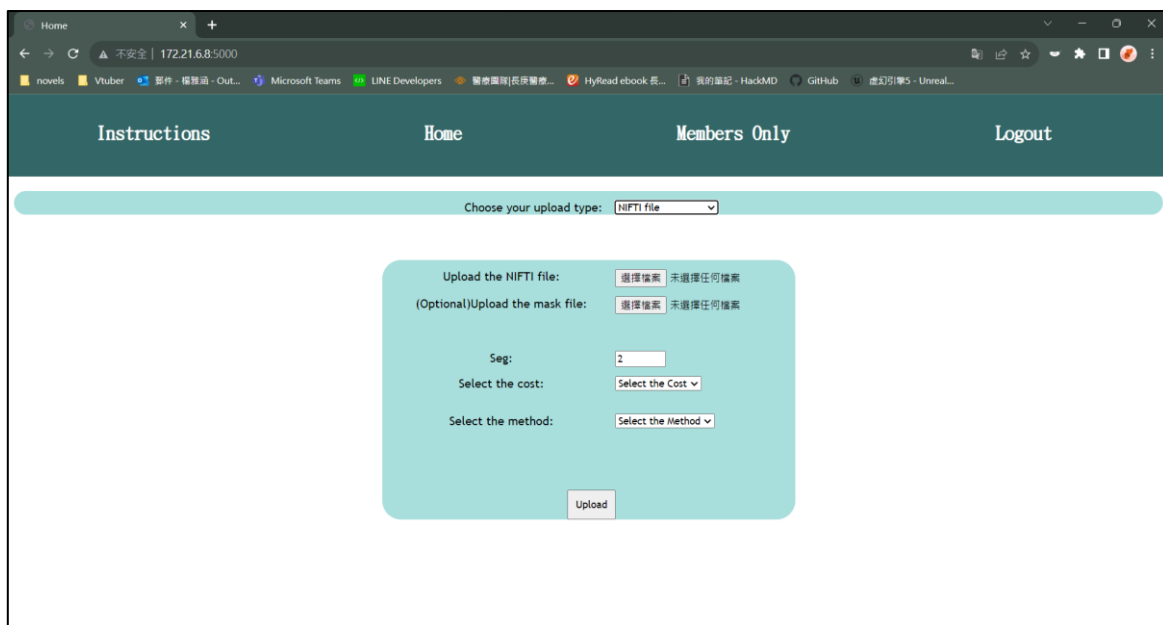


圖. NIFTI file

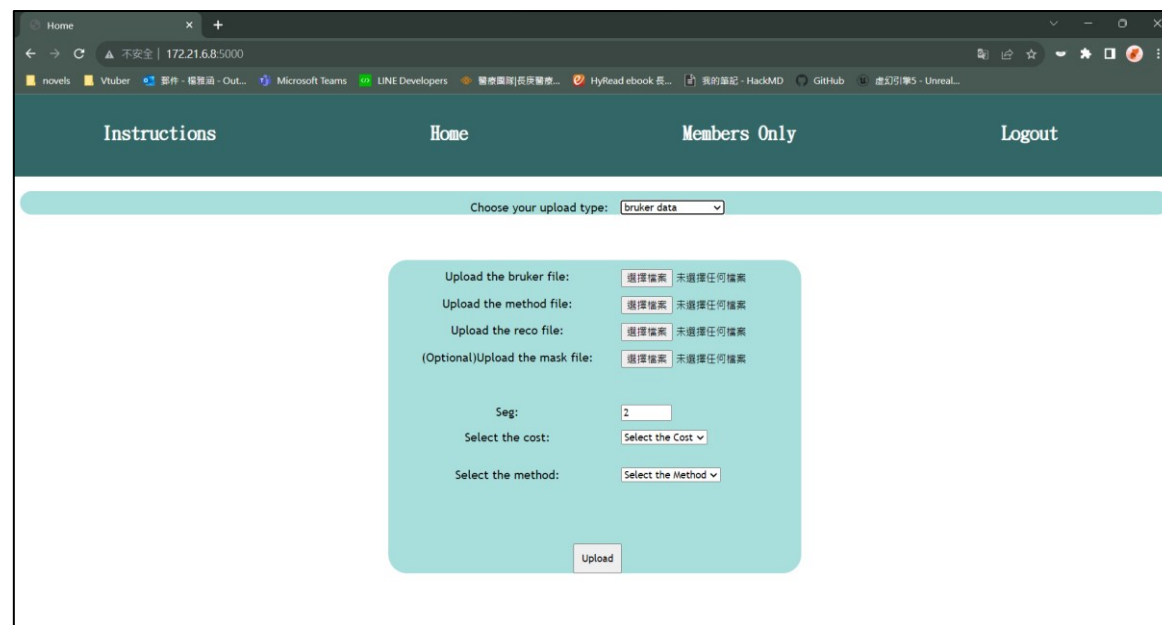


圖. bruker data

# 成果

選擇NIFTI檔案作為輸入。

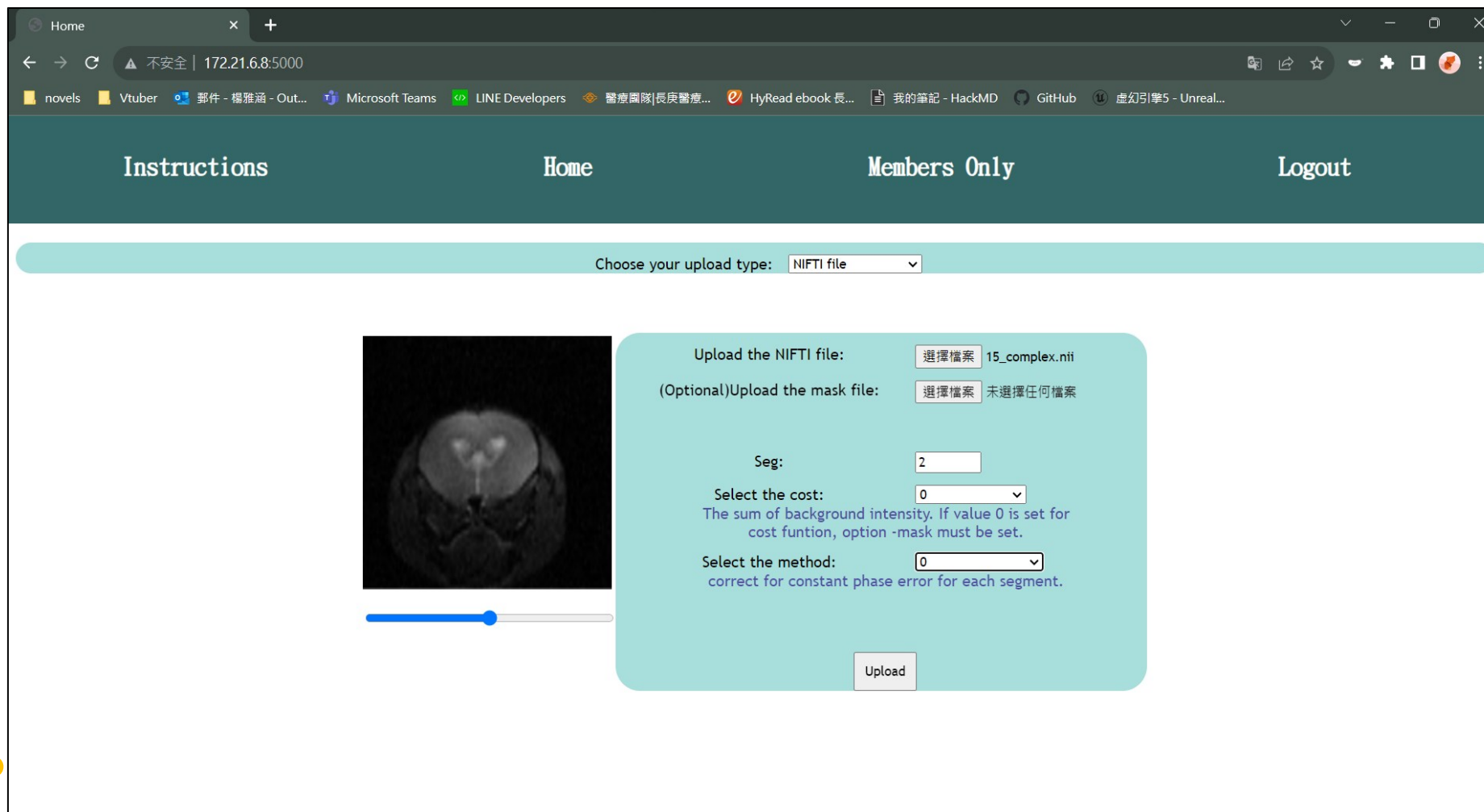
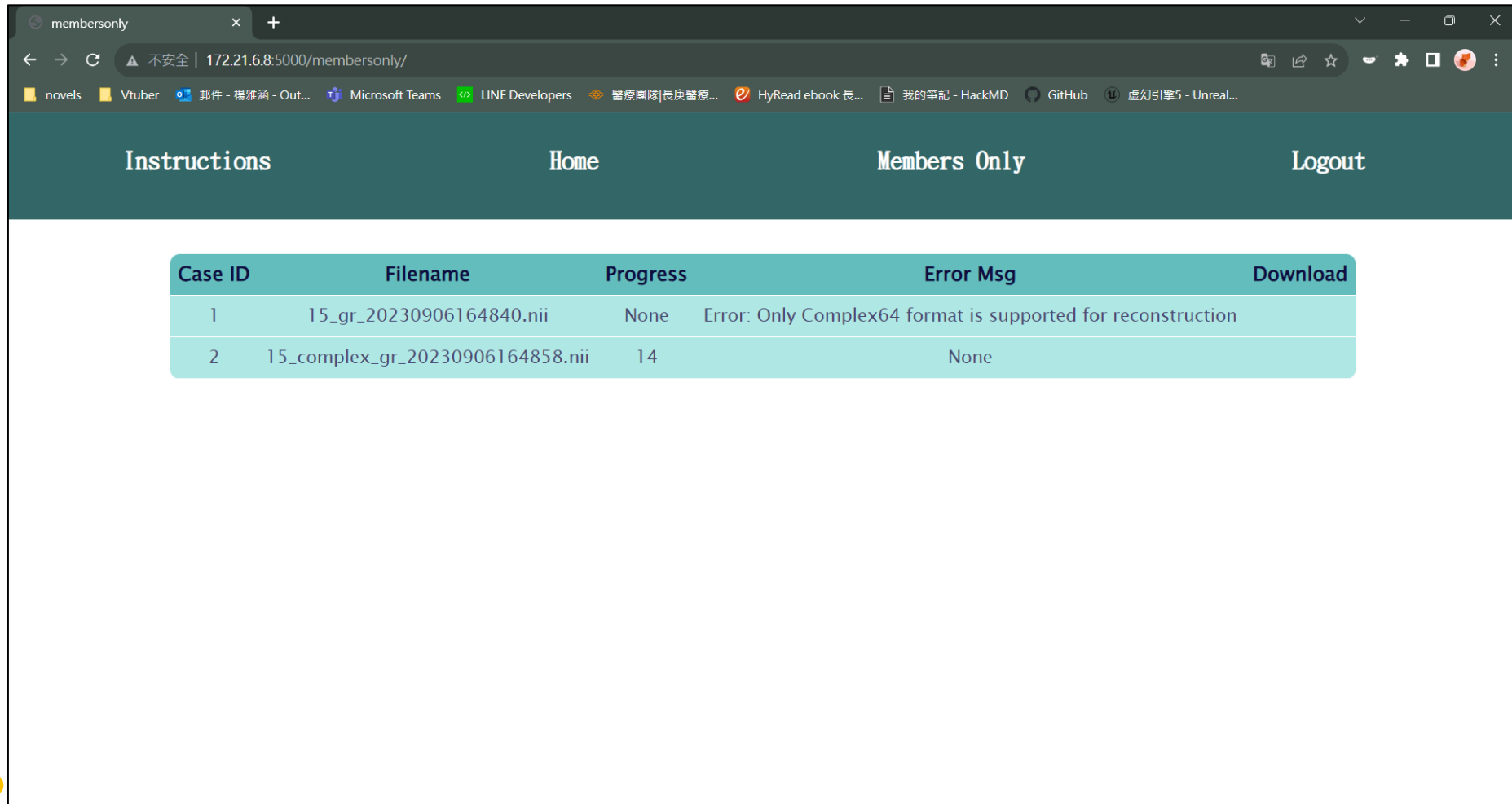


圖. 輸入NIFTI檔案

# 成果

系統會自動跳轉到會員專區，並呈現所有任務的狀態。

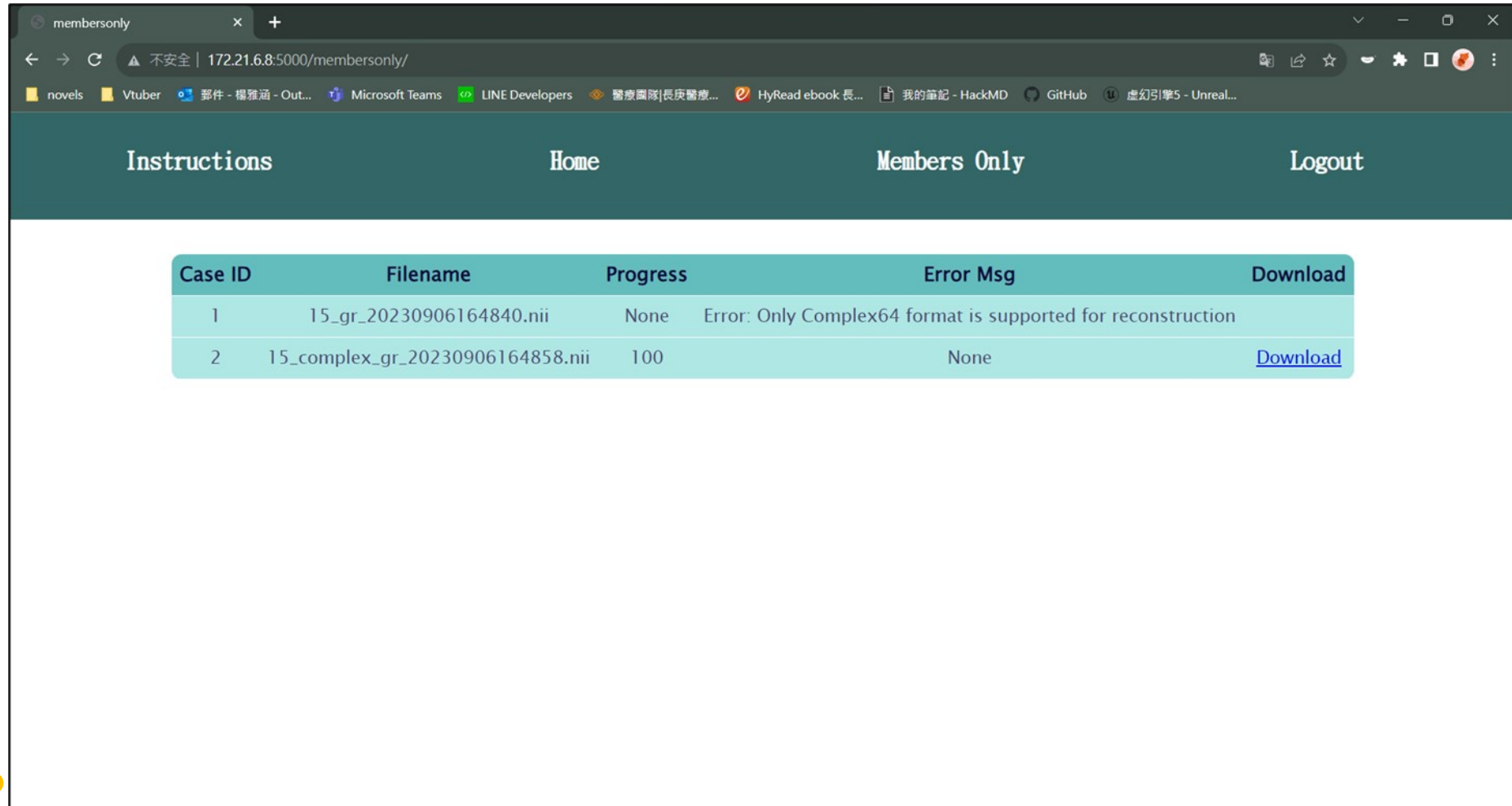


Case ID	Filename	Progress	Error Msg	Download
1	15_gr_20230906164840.nii	None	Error: Only Complex64 format is supported for reconstruction	
2	15_complex_gr_20230906164858.nii	14	None	

圖. 會員專區

# 成果

當任務進度為100時，會顯示下載連結，使用者可將結果儲存於本地端。



Case ID	Filename	Progress	Error Msg	Download
1	15_gr_20230906164840.nii	None	Error: Only Complex64 format is supported for reconstruction	
2	15_complex_gr_20230906164858.nii	100	None	<a href="#">Download</a>

圖. 會員專區

# 成果

導航欄有Instructions供使用者進行閱讀。

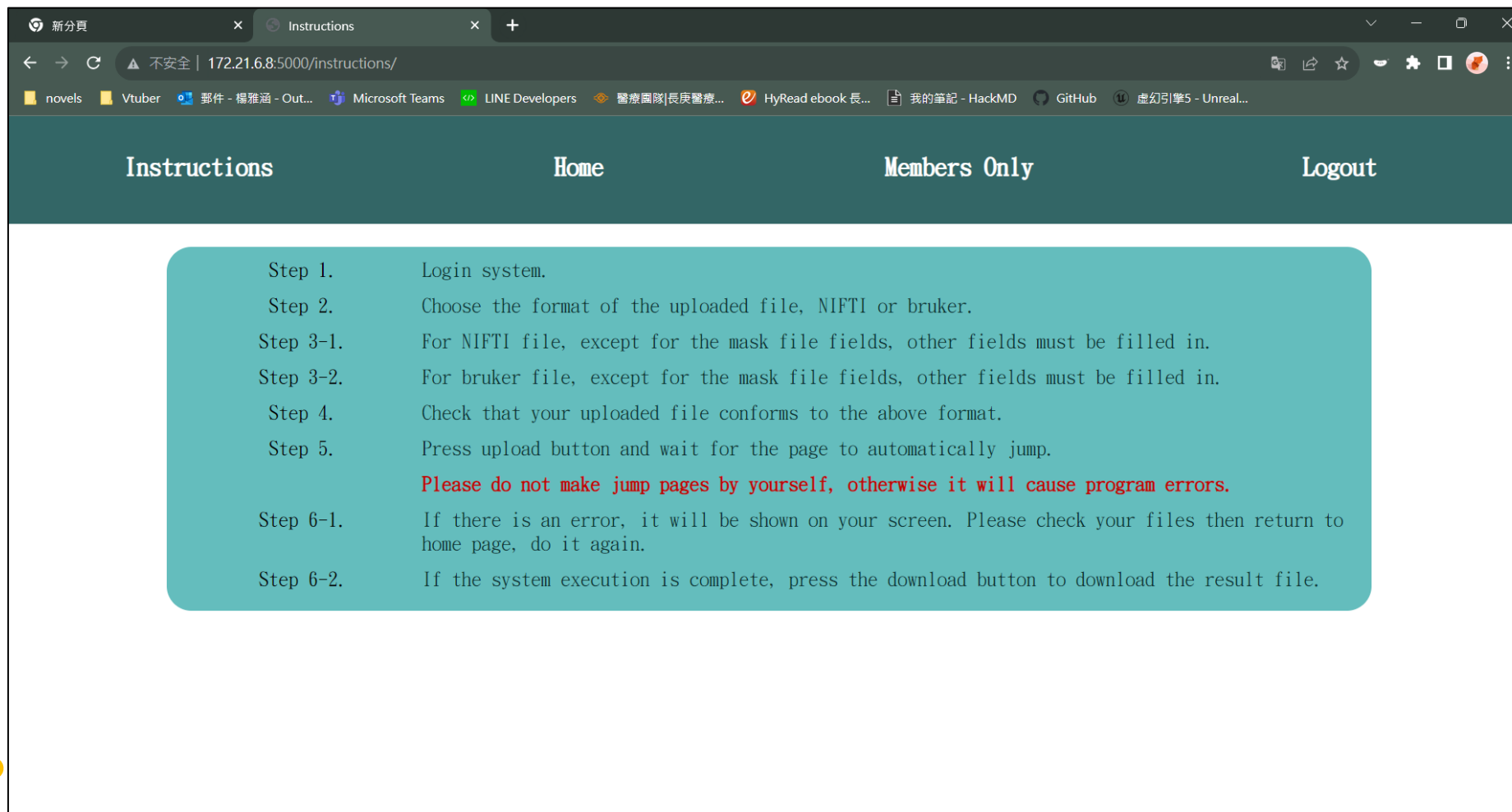


圖. 輸入NIFTI檔案