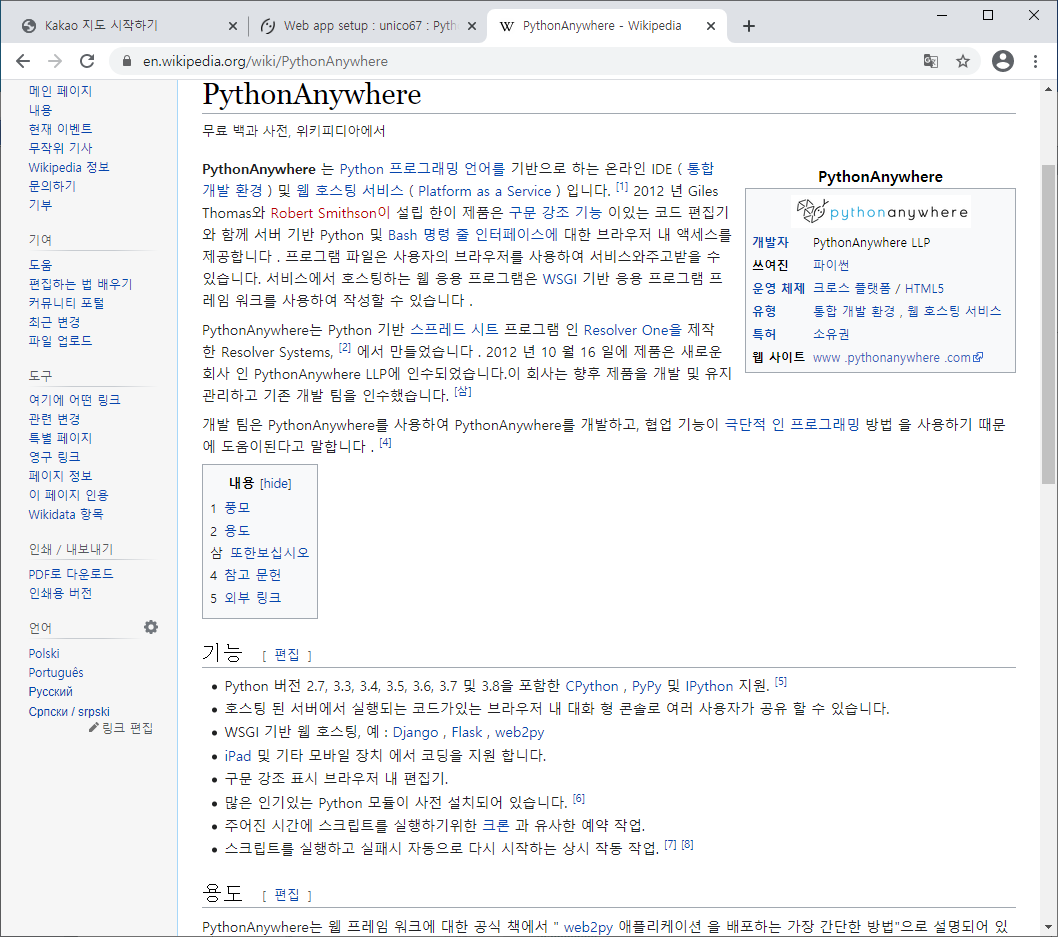
개발이 완료된 장고프로젝트는 현재 로컬에서만 테스트 가능하다.

우리가 팀프로젝트로 개발한 장고프로젝트를 언제든 그리고 어디서든 접속하여 페이지를 사용할 수 있도록

실제 웹 사이트에 배포하는 것이 여러모로 좋을 것이다.

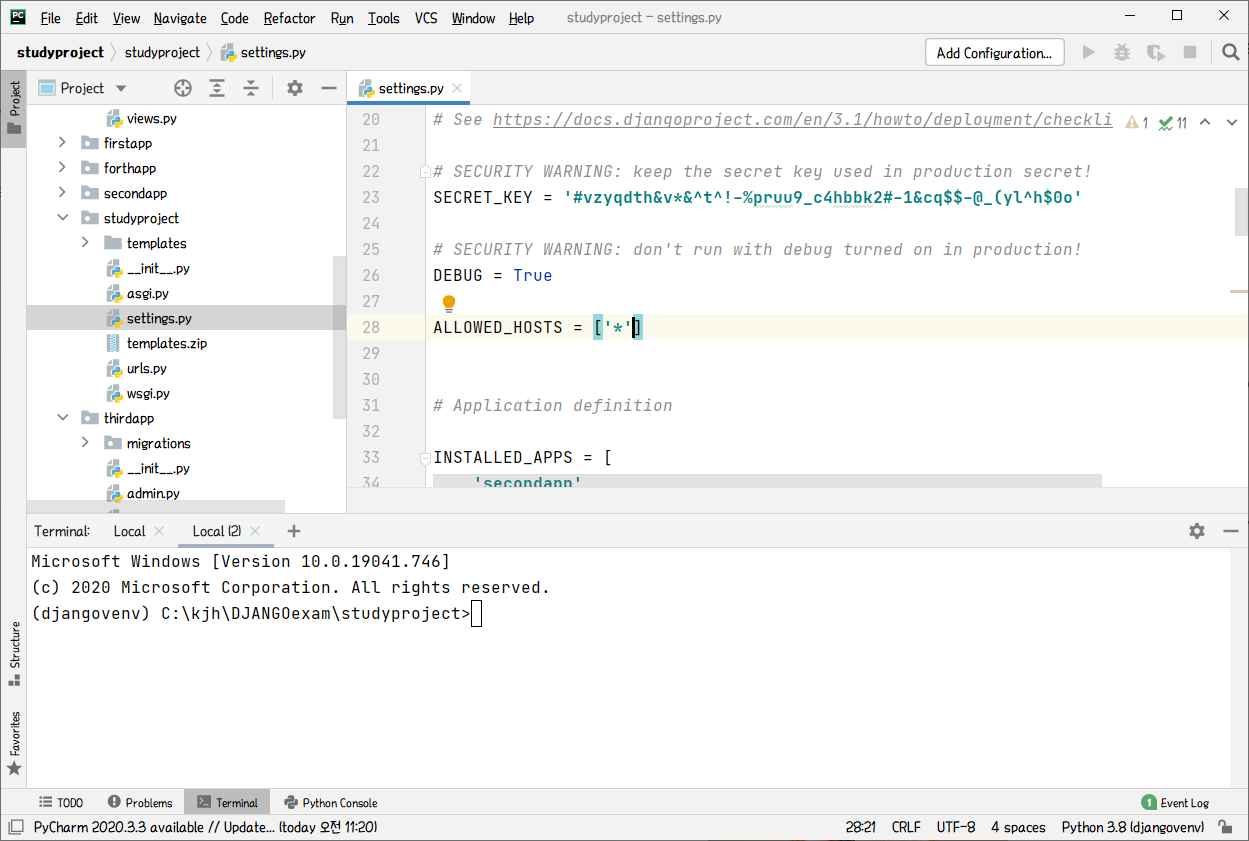
이렇게 개발이 완료된 웹 프로젝트의 결과물 즉 웹 애플리케이션을 웹상에서 서비스해 주는 기술을 웹 호스팅 서비스라고 하며 파이썬으로 구현한 웹 애플리케이션은 pythonanywhere 사이트에 올려서(배포해서) 서비스하는 것을 추천한다.



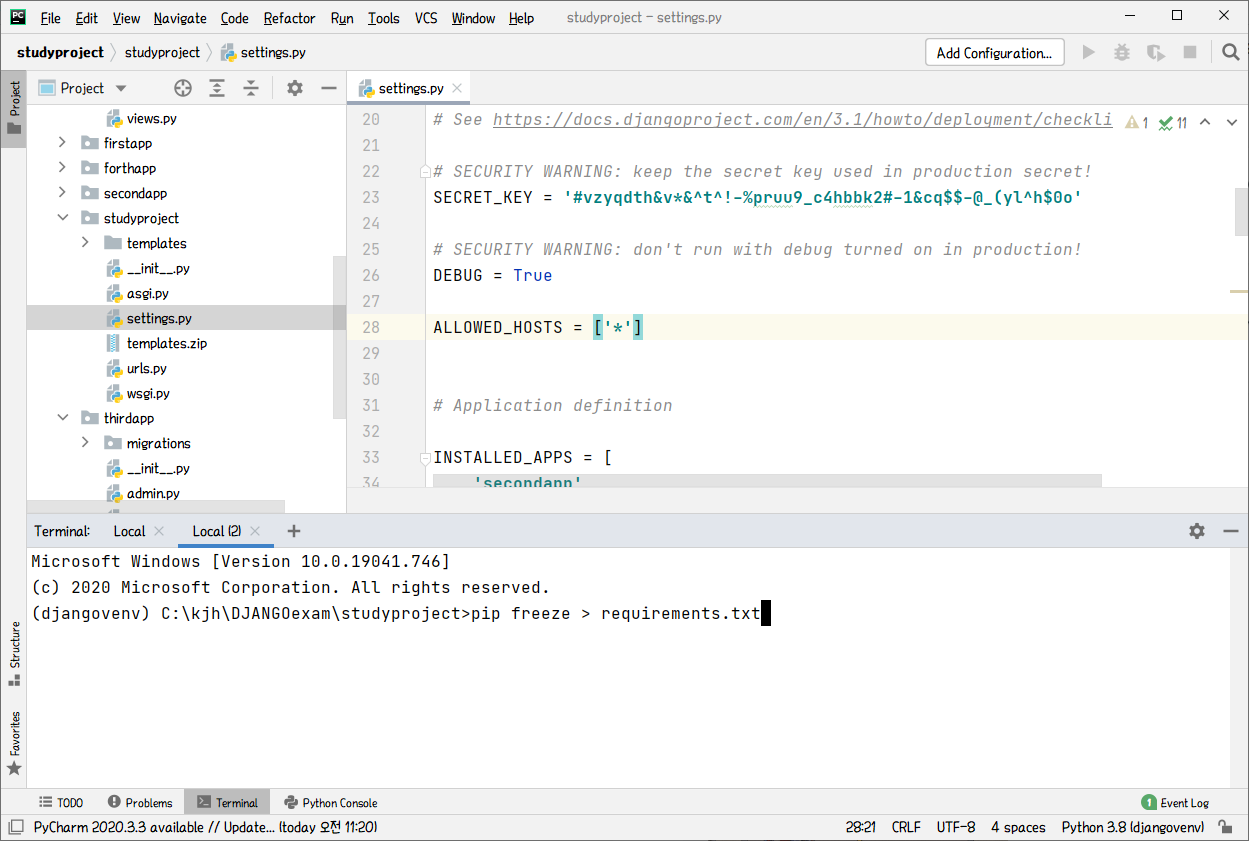
pythonanywhere 사이트에 올리기(배포) 위해서는 다음에 제시된 과정을 진행한다.

(1) 먼저 프로젝트폴더명/settings.py 파일에서 ALLOWED\_HOSTS 변수의 값을 [ ] 에서 [‘\*’] 로 변경한다.

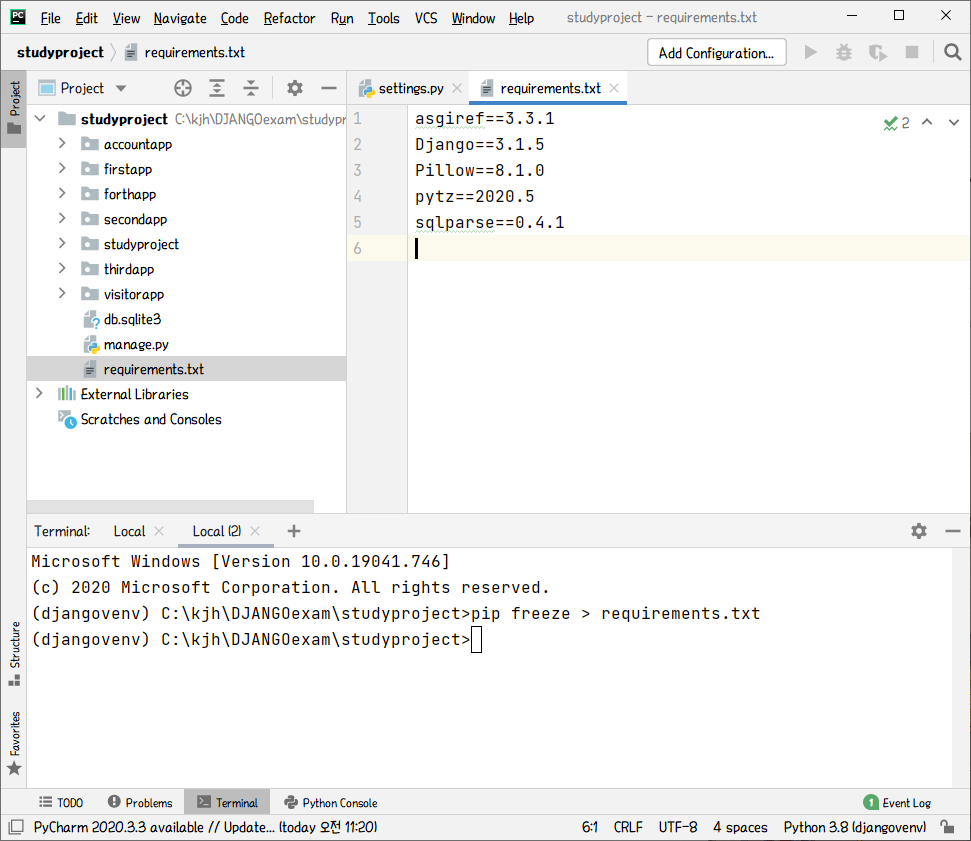
([ ]은 localhost 로만 접근 가능한 웹 애플리케이션을 생성하게 된다.)



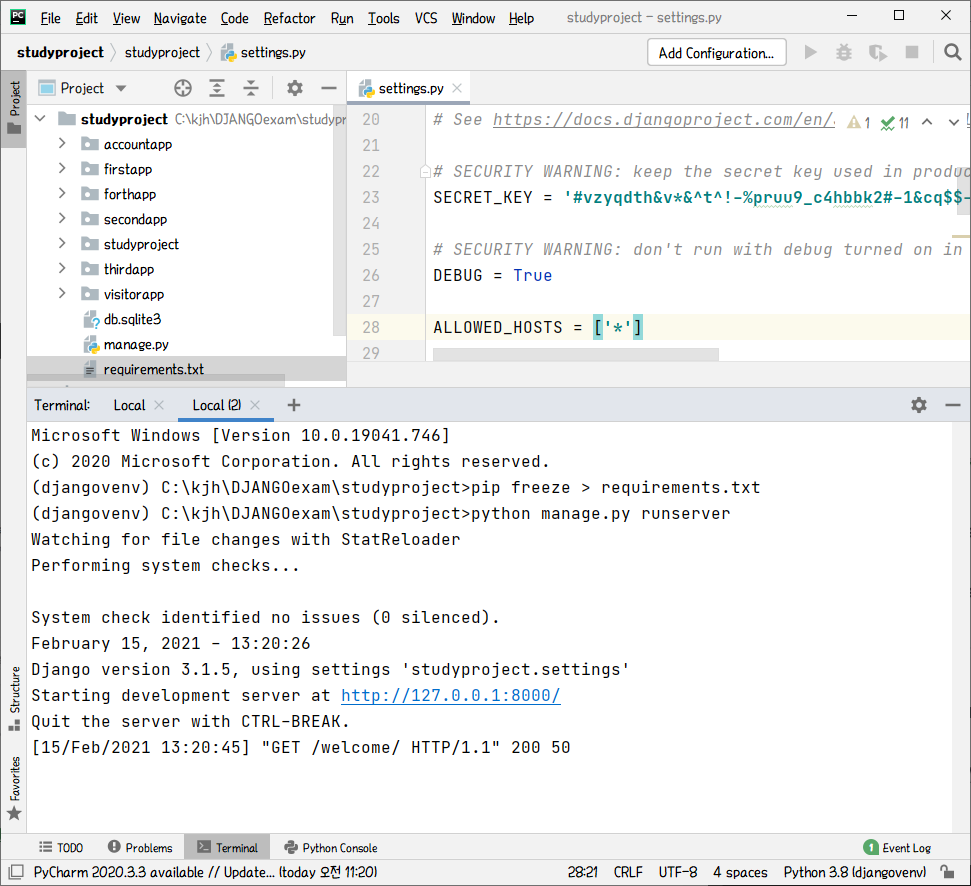
(2) 터미날 창에서 다음 명령을 수행시켜서 설치된 패키지 정보를 저장하는 requirements.txt 를 생성한다.

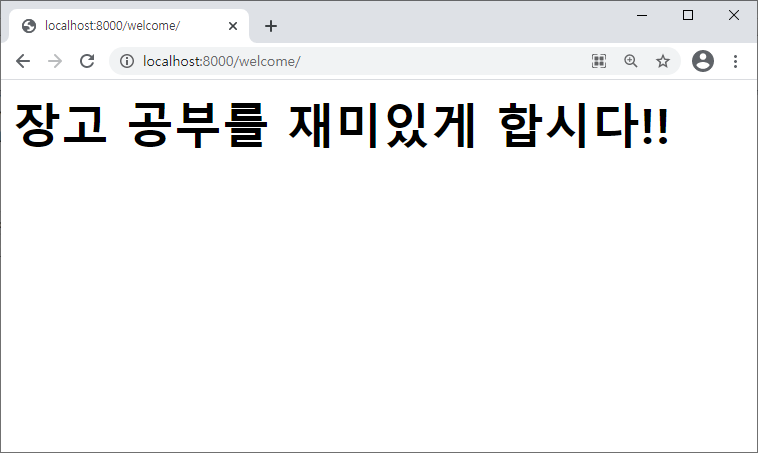


그러면 다음과 같은 내용으로 구성되는 requirements.txt 파일이 생성된다.

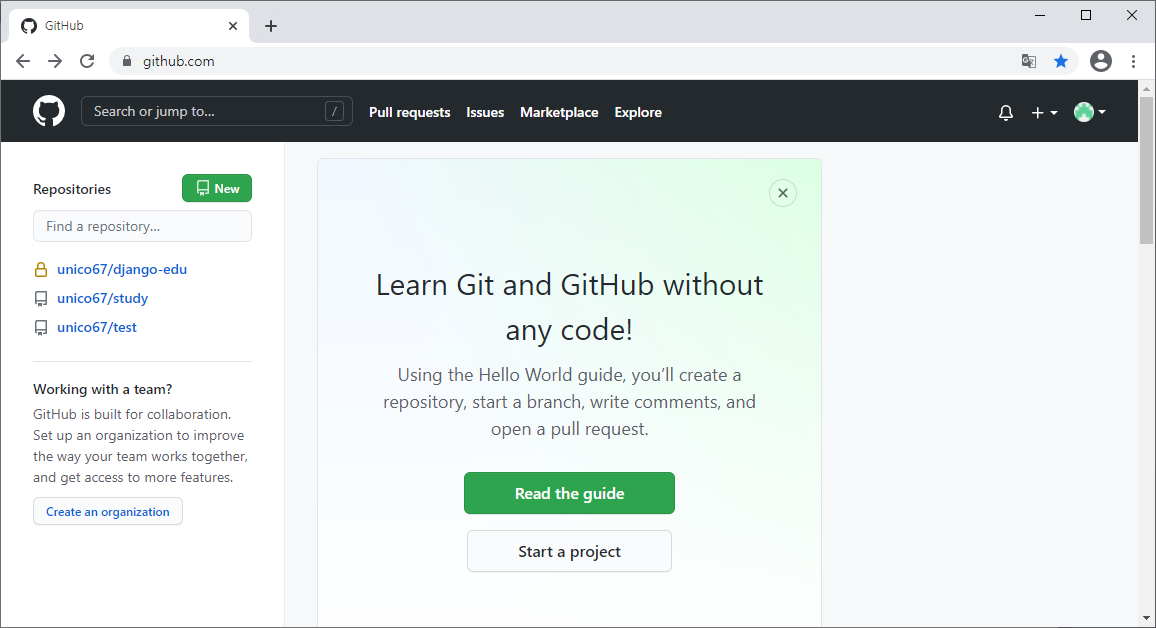


그런 다음 다음과 같이 장고서버를 기동시키고 웹 페이지 하나를 요청해서 제대로 결과가 나오는지 확인한다.

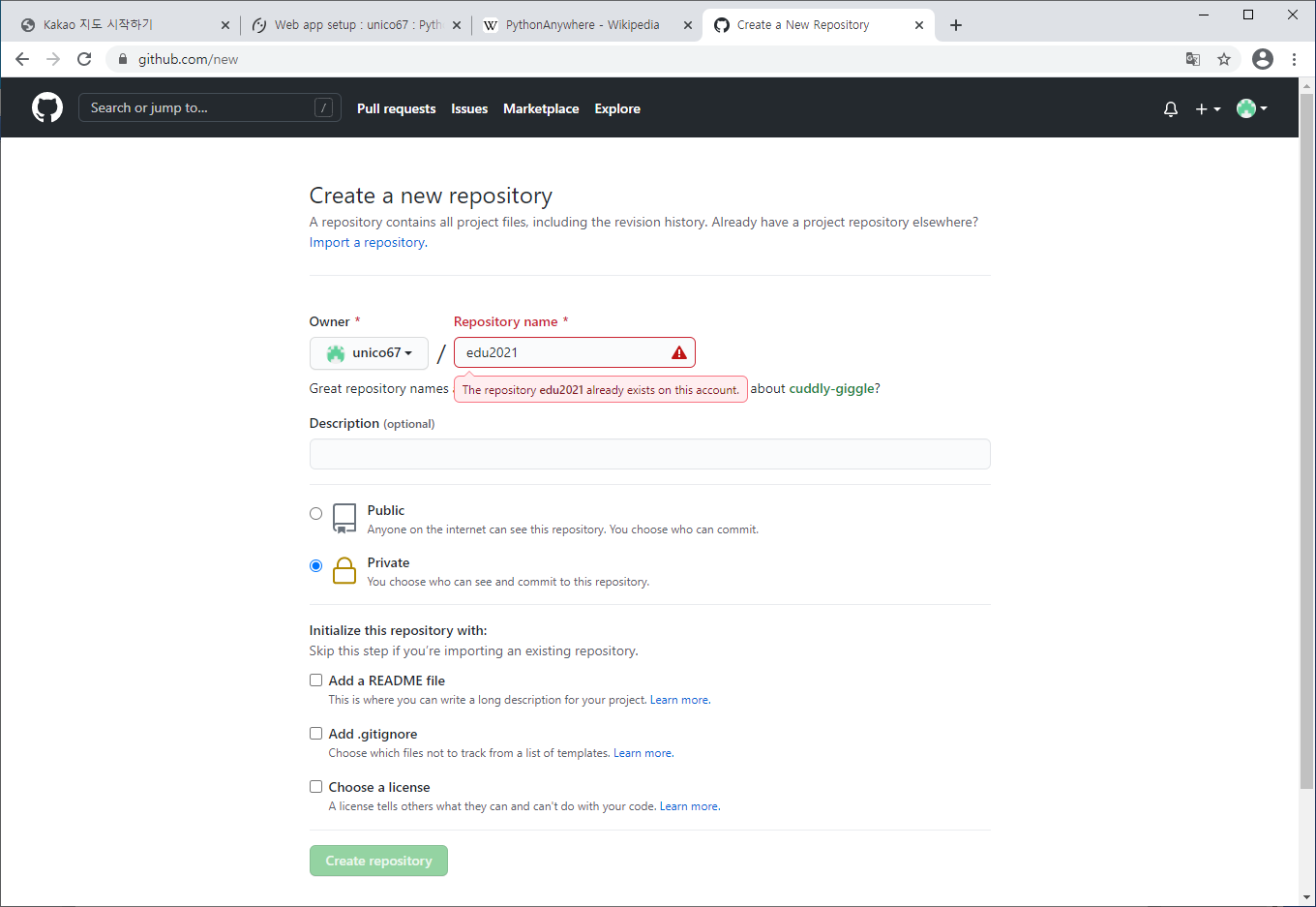




(3) 깃허브에 로그인 하고 새로운 저장소를 한 개 만든다.

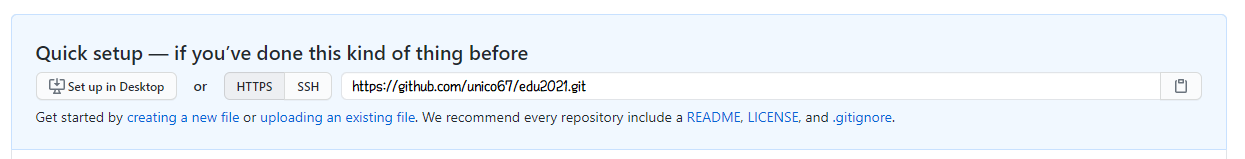


강사는 다음과 같이 edu2021 이라는 명칭으로 저장소(repository)를 생성했다.

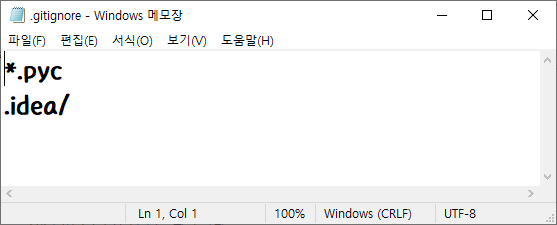


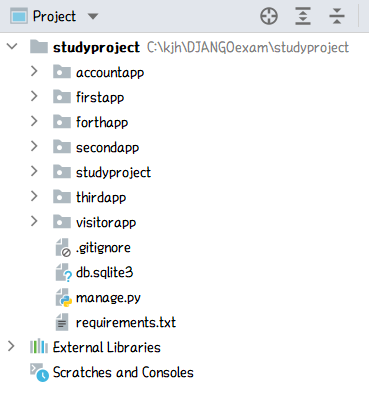
다음 화면에 출력된 URL 을 메모장에 보관한다.

https://github.com/unico67/edu2021.git



(4) 깃허브에 새로이 생성한 저장소(강사는 edu2021)에 프로젝트 소스를 업로드하기 위해서 다음과 같은 내용으로 .gitignore 라는 파일을 프로젝트 폴더에 생성한다. 이 파일의 용도는 업로드하지 않으려는 파일들에 대한 정보를 작성하는 것이다.

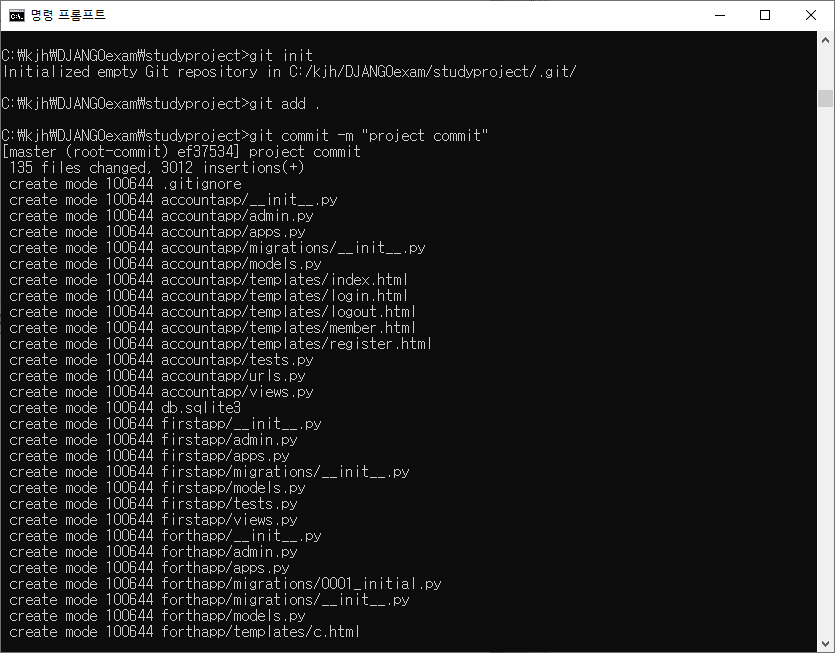




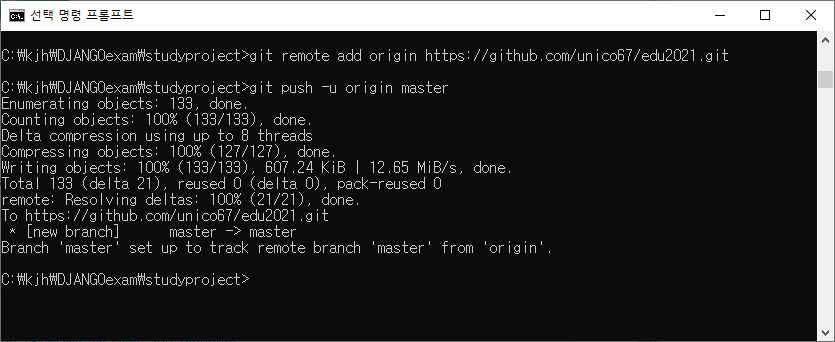
(5) 명령 프롬프트(cmd) 창을 기동시키고 다음 명령들을 수행시킨다.

다음 명령을 실행시켜서 프로젝트 폴더로 이동한다.

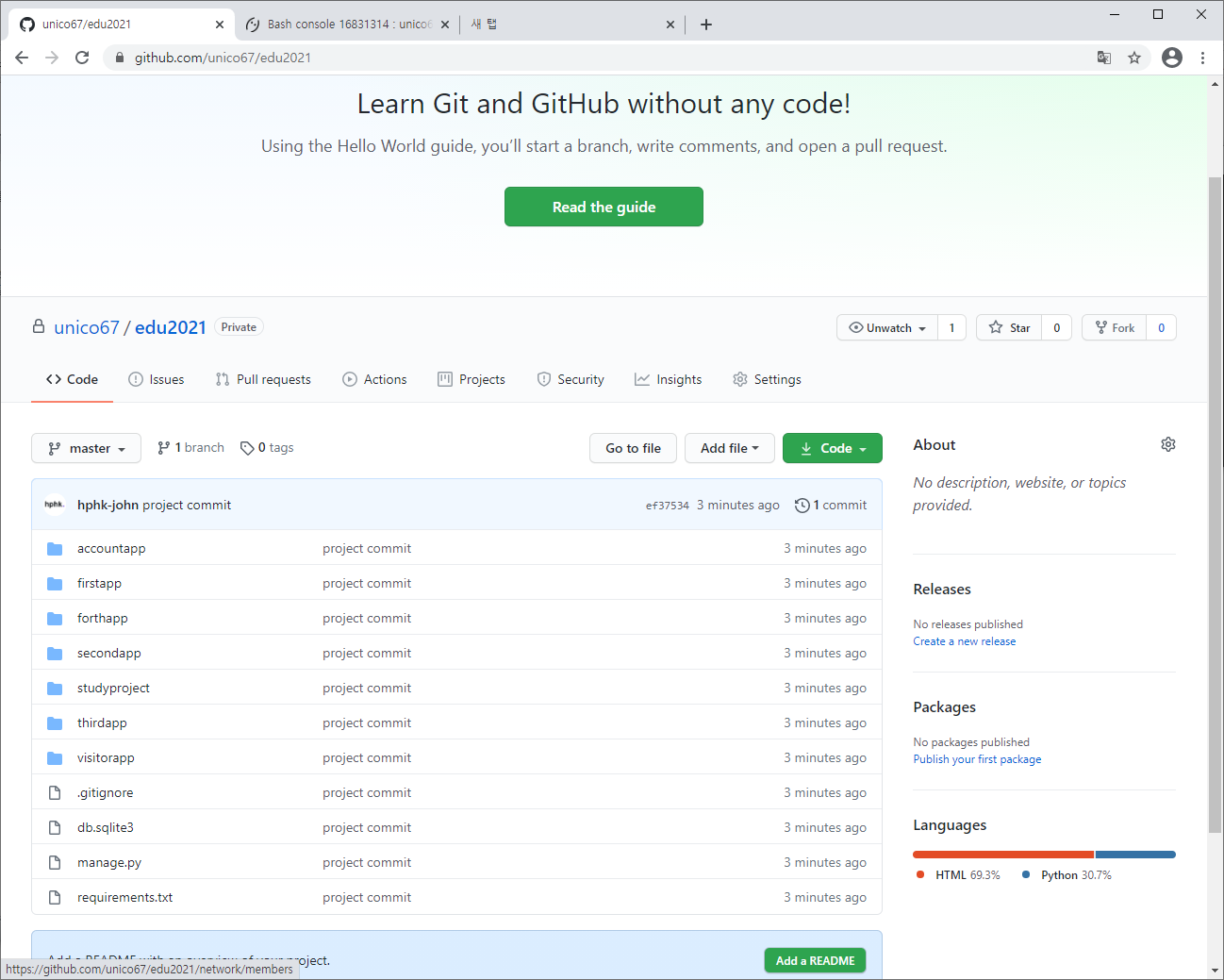
(cd c:\kjh\DJANGOexam\studyproject)



**본인꺼로**



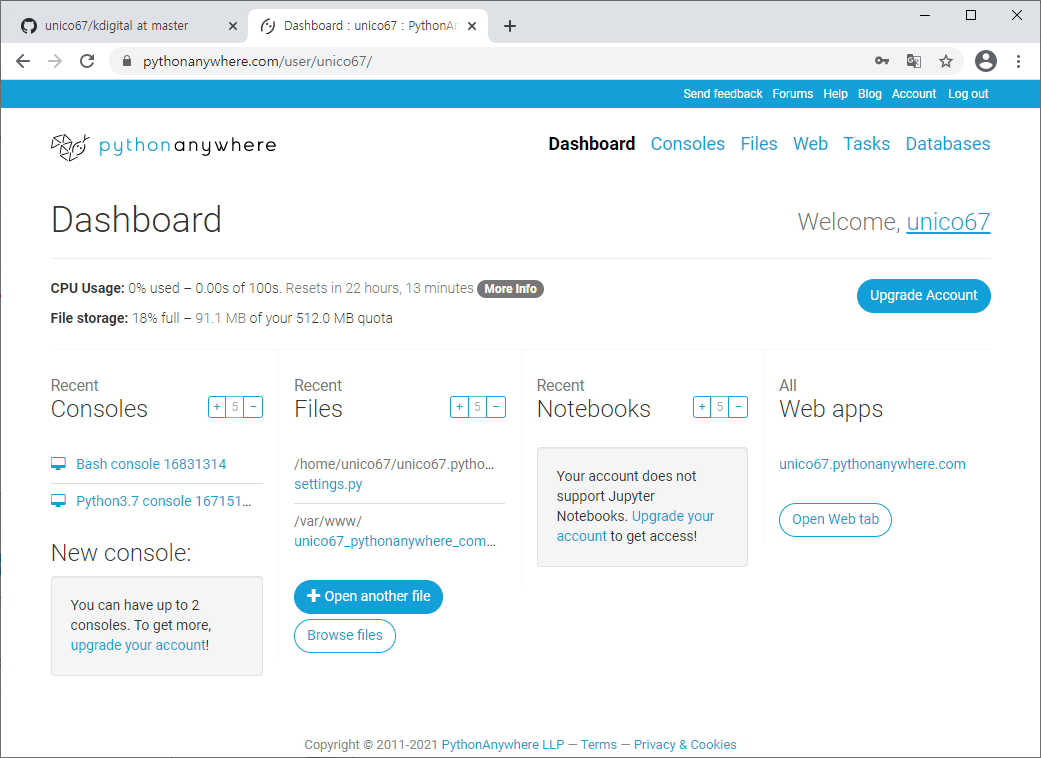
다음과 같이 edu2021 저장소에 파일들이 저장된 것을 확인할 수 있다.



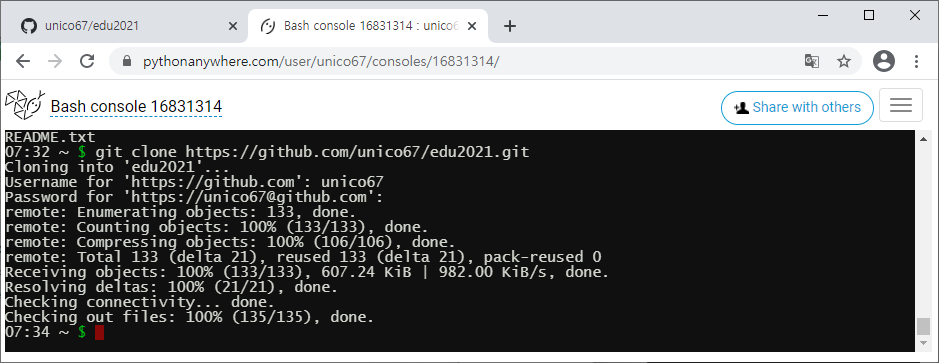
여러분들은 이미 깃허브를 사용하고 있었으므로 (3),(4),(5)은 스킵해도 될거 같네용…….

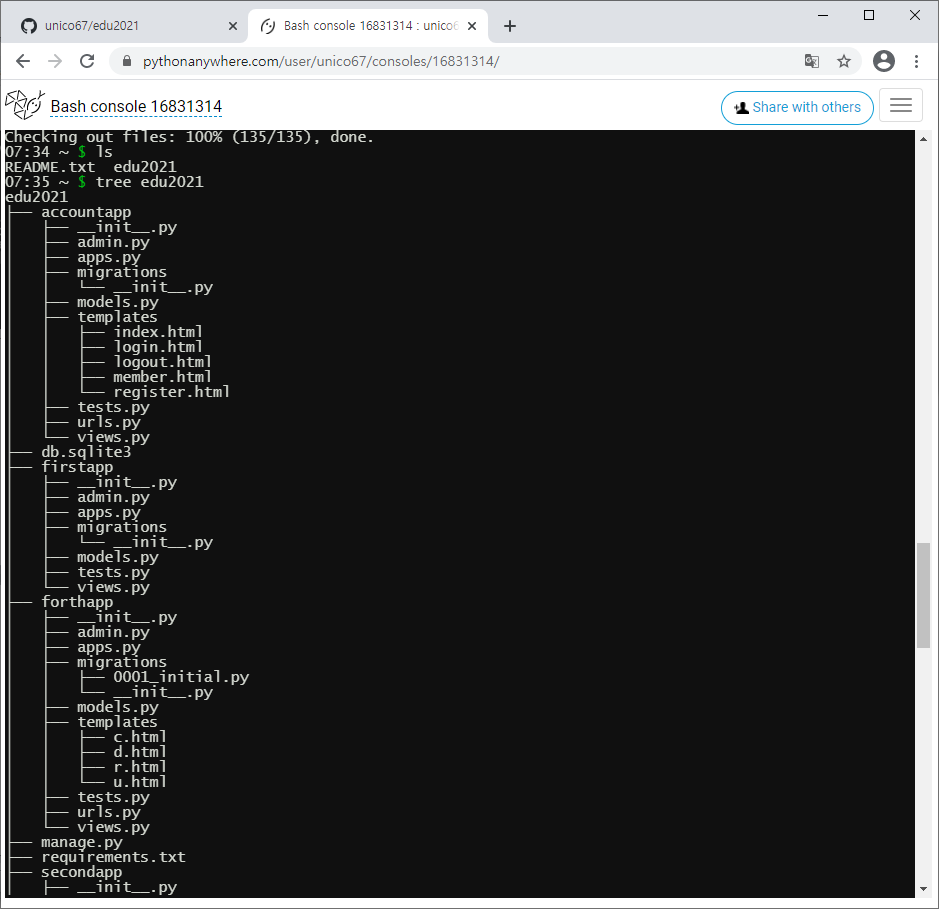
(6) pythonanywhere 사이트에 회원 가입해서 계정을 하나 만든다. Dashboard 화면에서 Bash를 클릭하여

명령프롬프트 창과 비슷한 화면을 기동시킨다.

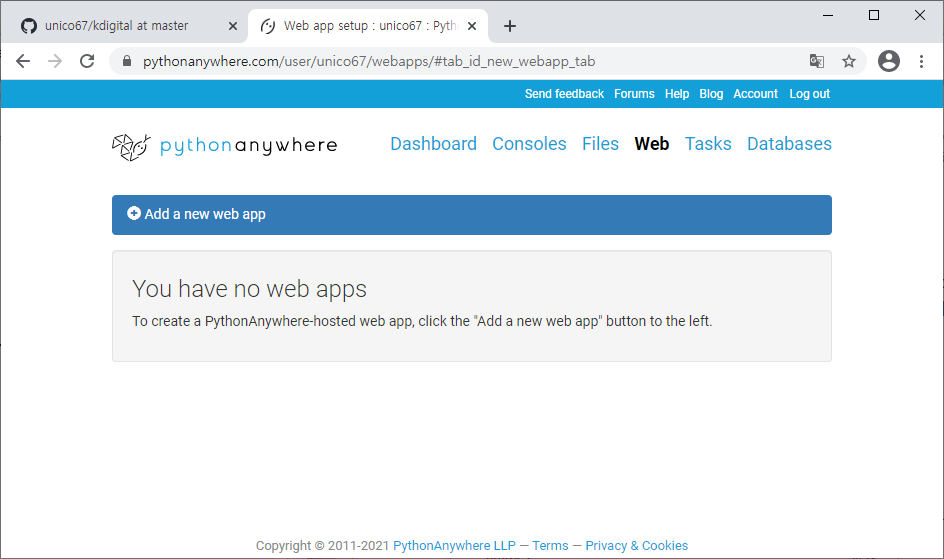


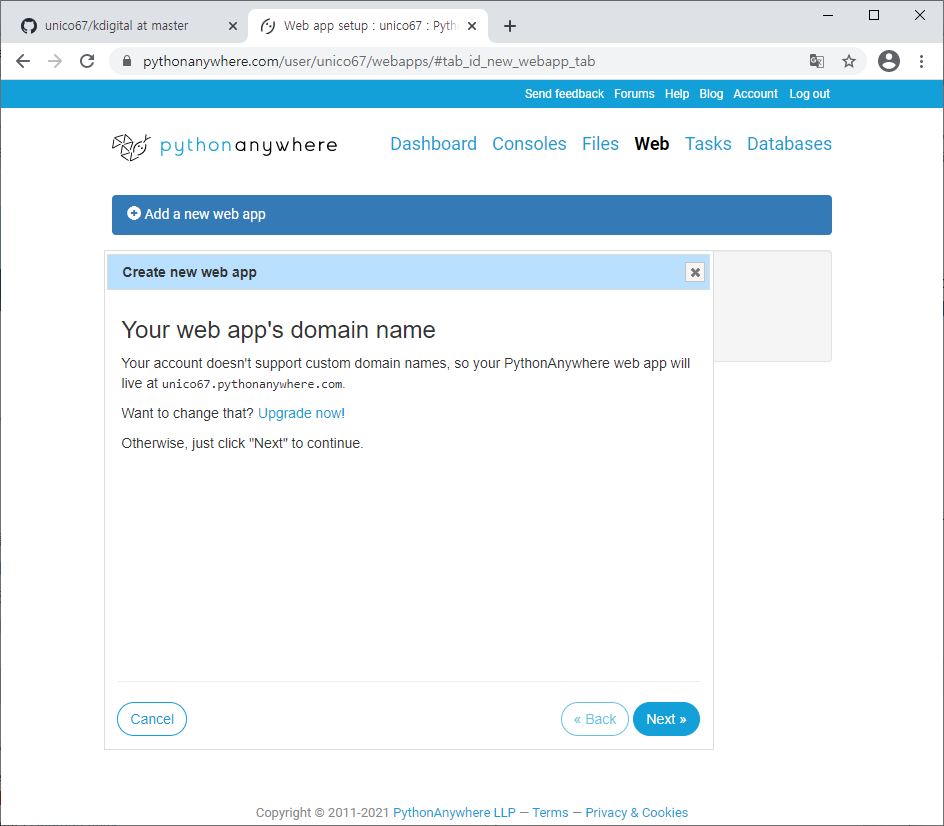
다음 명령을 실행시켜서 깃허브에 있는 edu2021 저장소의 파일들을 복제해 온다.

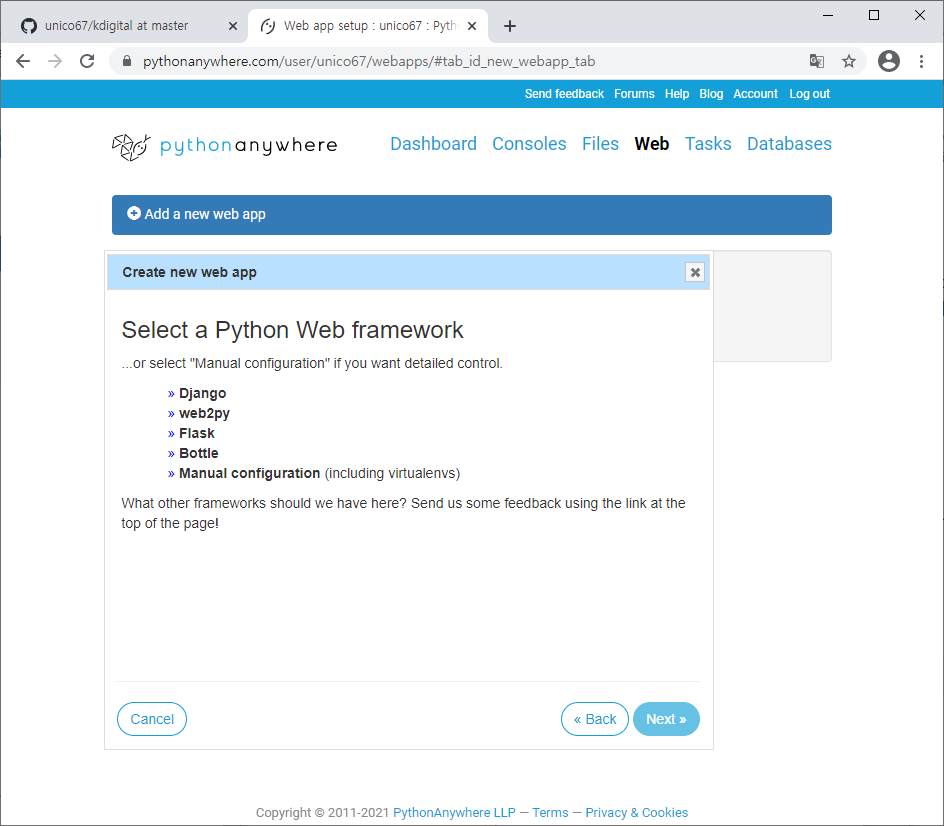


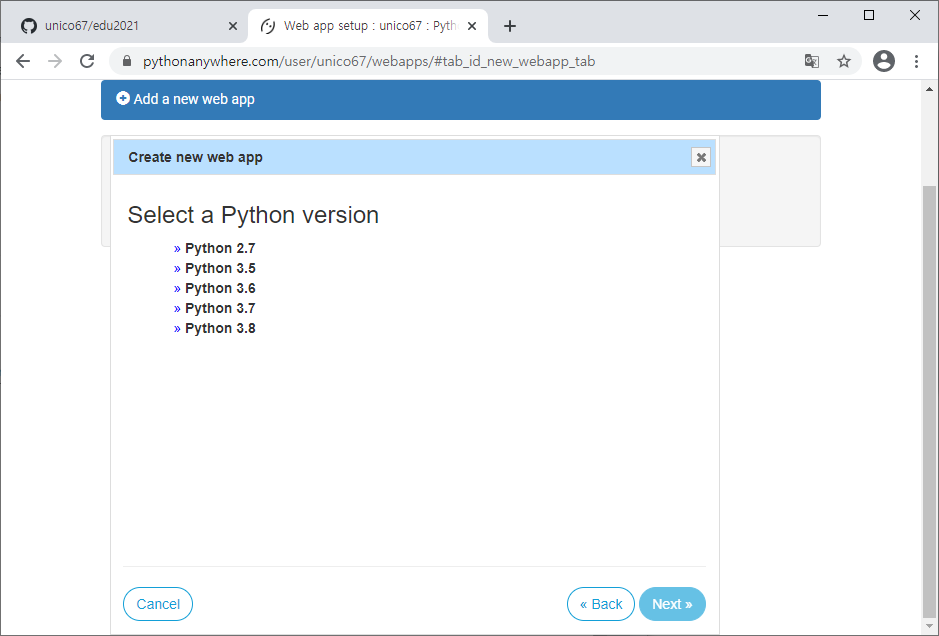


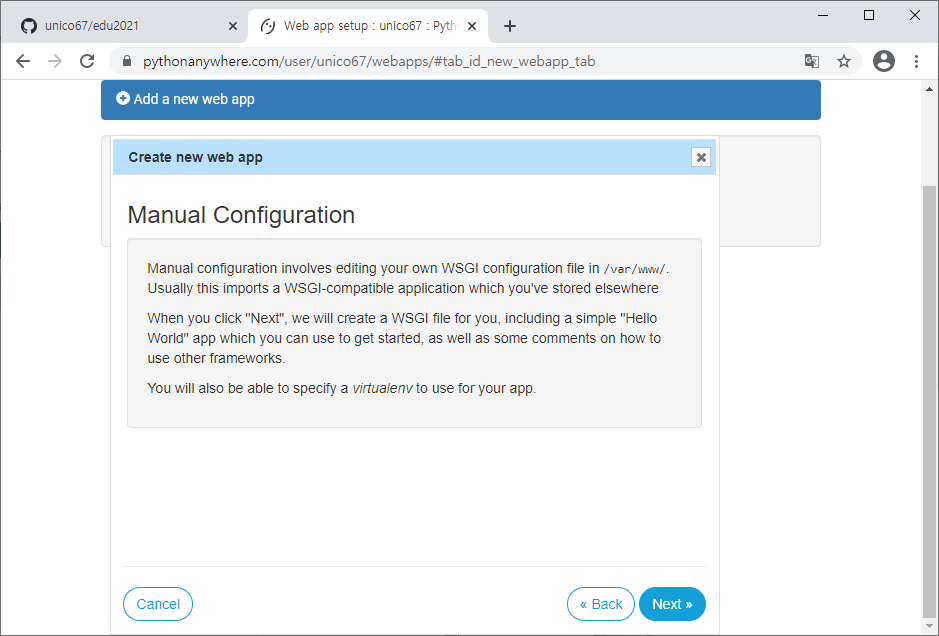
(7) pythonanywhere 사이트에 새로운 웹앱을 생성한다.





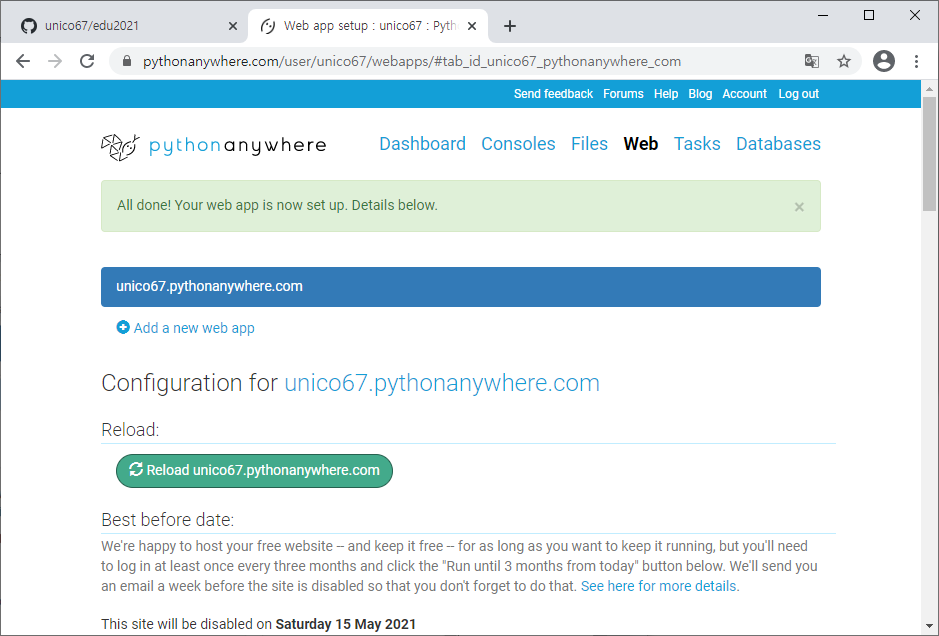




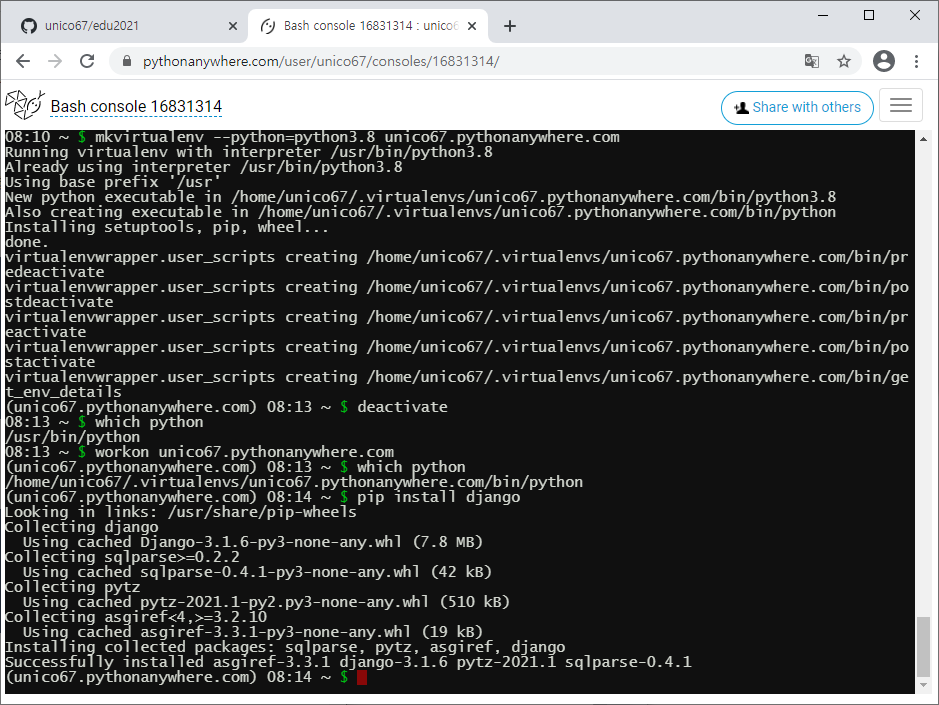


다음과 같은 웹앱 설정화면이 출력된다.

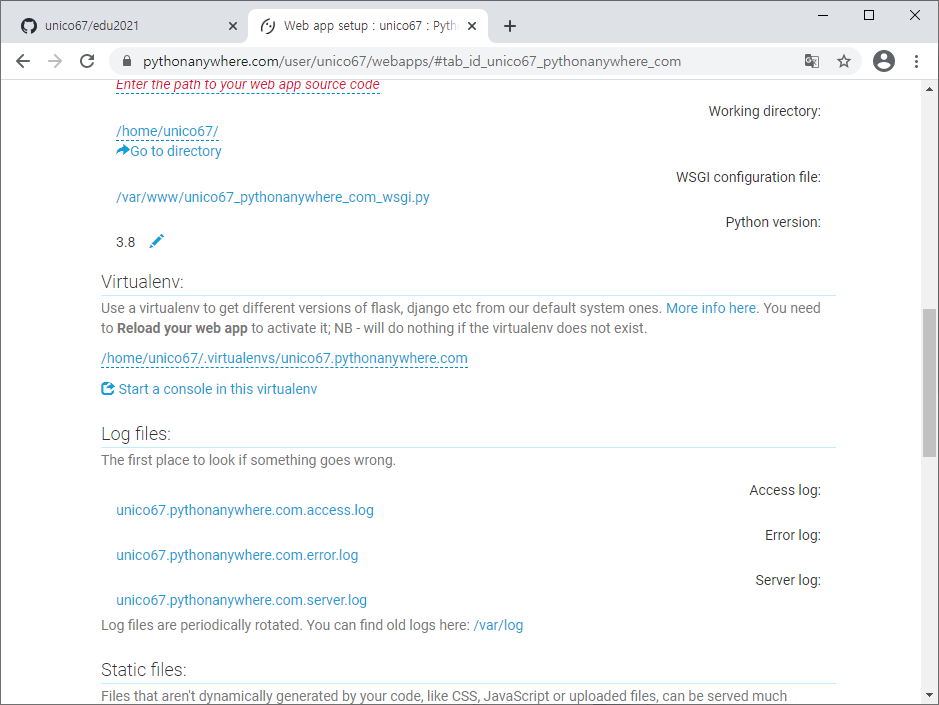
앞으로의 작업은 대부분 이 화면에서 환경 설정을 처리하게 된다. 수정을 모두 끝낸 후에는 Reload…… 버튼을 시스템에 클릭하여 반영한다.



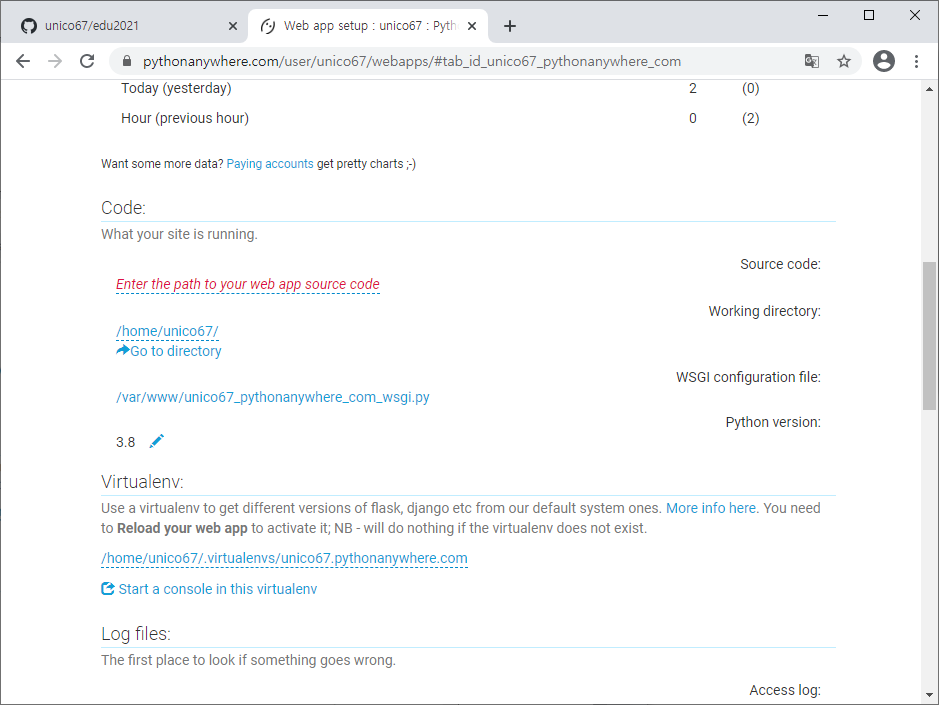
(8) 복제해온 장고프로젝트를 위한 가상환경을 생성하고 여기에 django 캐피지를 설치한다.



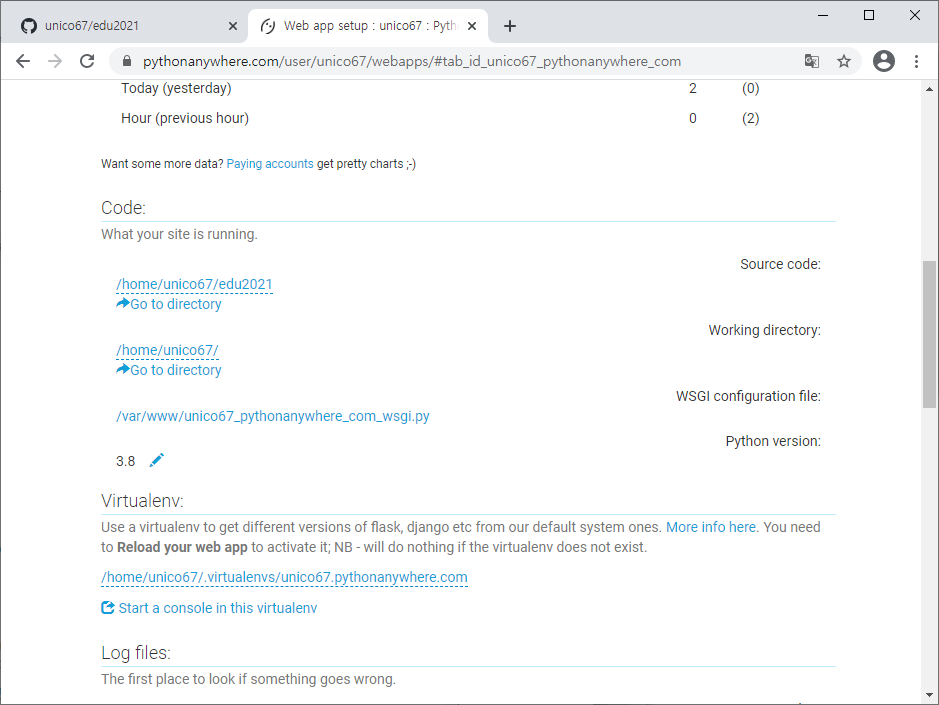
웹앱 화면에서 Virtualenv 항목에 가서 다음과 같이 작성하여 가상환경 패스를 등록한다.



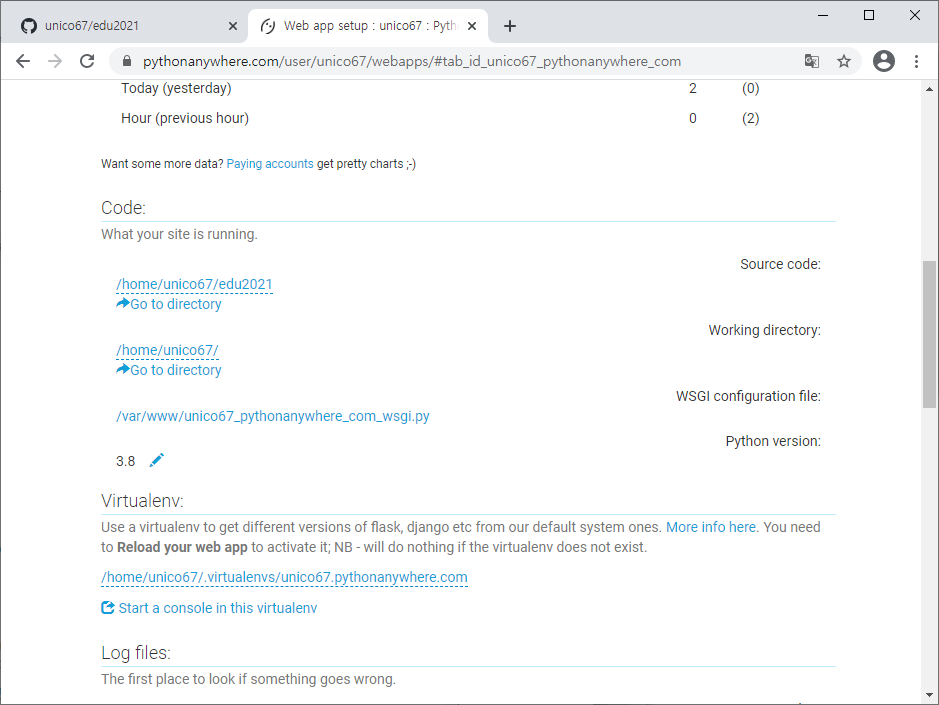
(9) 소스들이 존재하는 폴더를 설정하기 위해서 Code 항목에 있는 Enter the…… 링크를 클릭한다.



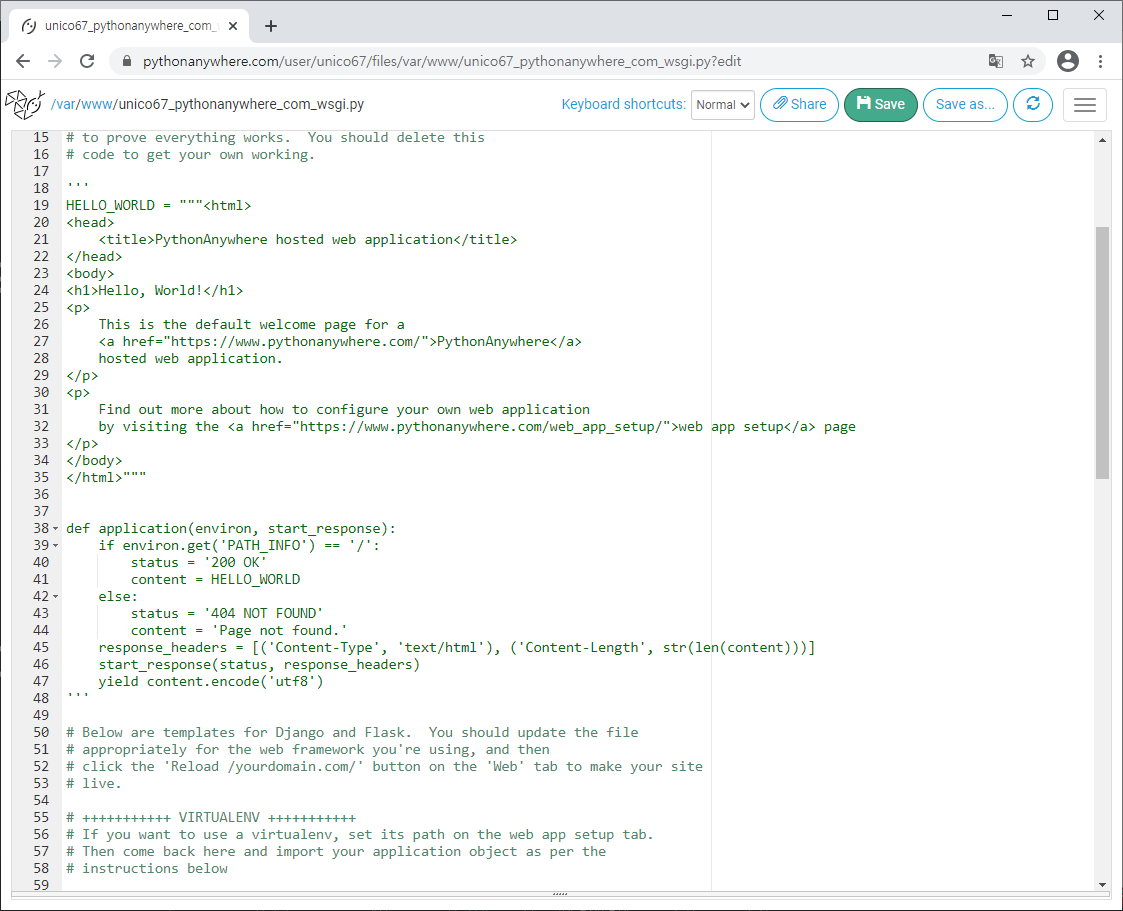
다음과 같이 소스 폴더의 패스를 작성한다.



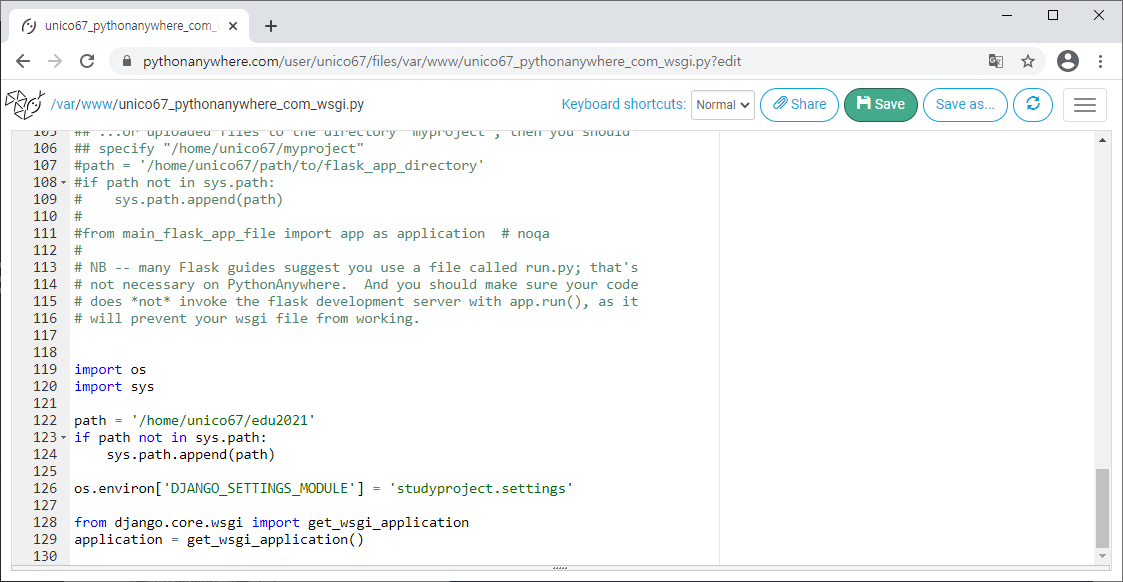
(10) WSGI configuration file 을 수정하기 위해서 다음 링크를 클릭한다.



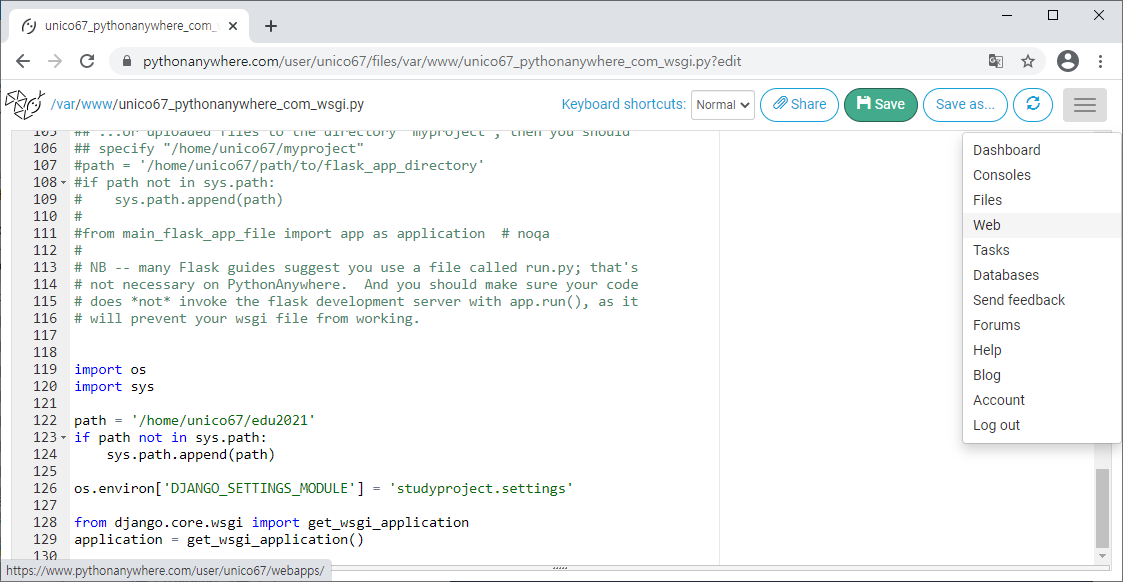
다음과 같이 에디터 창에 파일 내용이 출력되는데 19행부터 47행까지를 주석으로 묶기 위해 ‘’’ 를 입력한다.



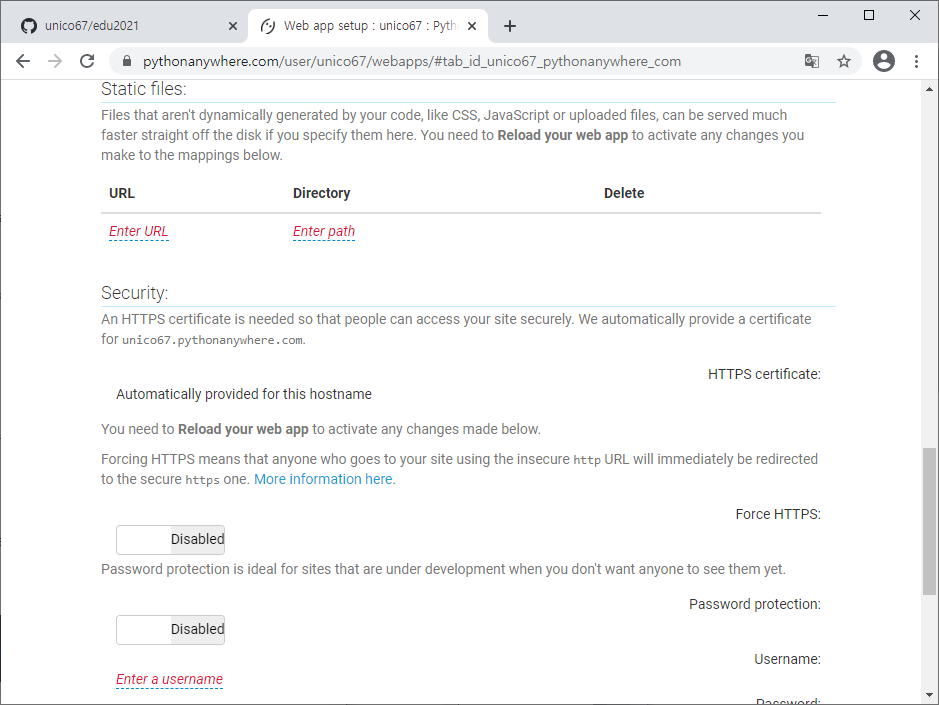
마지막 부분에 다음에 제시된 내용을 작성하고 Save 버튼을 클릭한다.



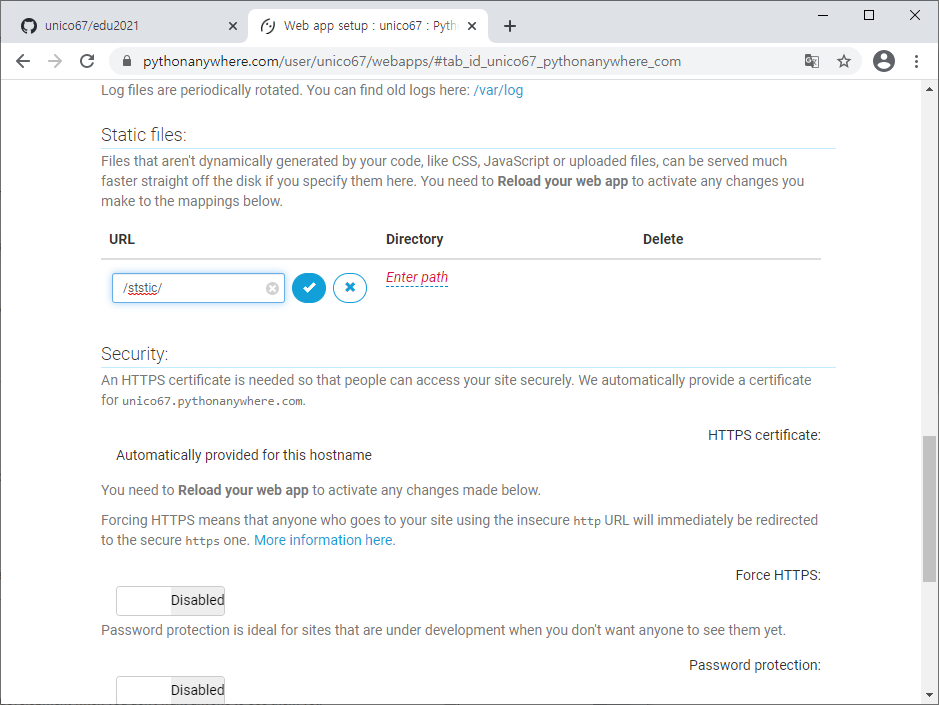
Save 버튼을 클릭한 다음에는 다음과 같이 버튼과 메뉴를 클릭하여 웹 화면으로 나간다.



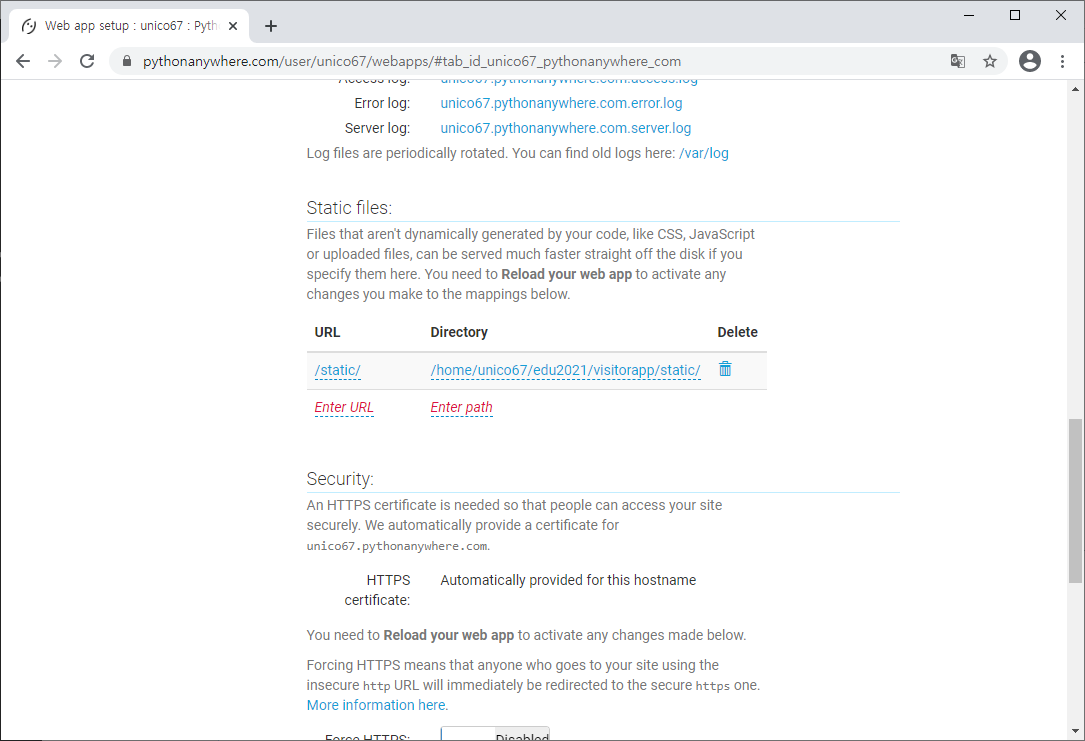
(11) Static files 를 설정하여 ststic 자원들이 존재하는 디렉토리 정보를 조정한다.



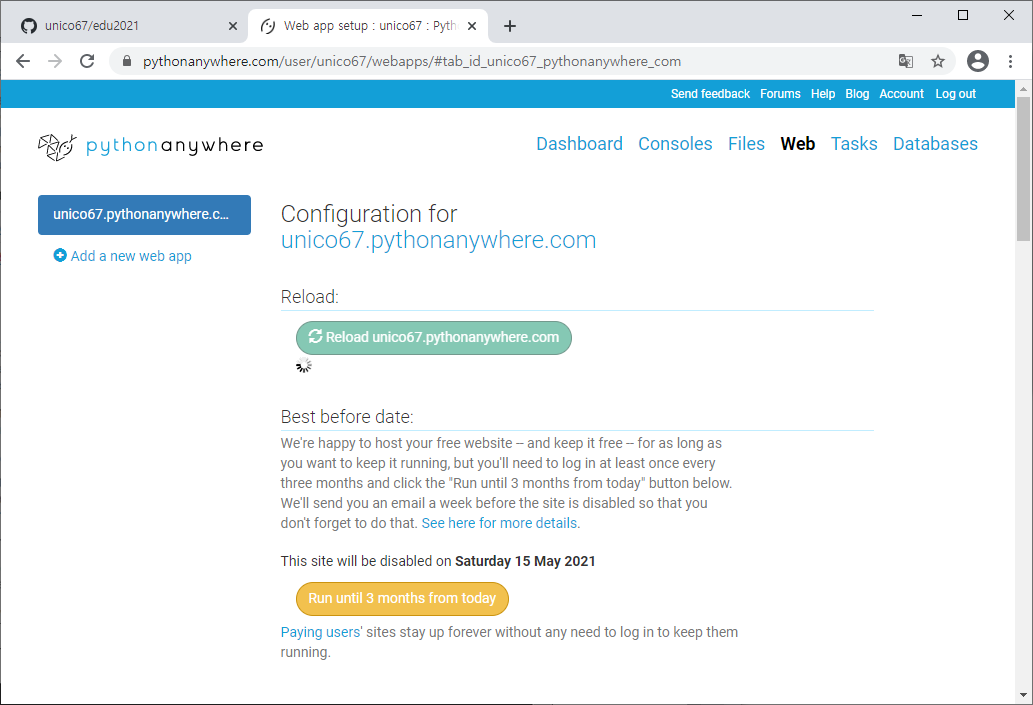
URL 에는 /static/ 을 입력하고 파란색 채크 이미지를 클릭한다.



Directory 에는 static 폴더에 대한 패스를 입력하고 파란색 채크 이미지를 클릭한다.



지금까지의 모든 변경을 반영하도록 윗쪽에 있는 Reload…… 버튼을 클릭한다.



부족한 설명은 다음 사이트 참고

https://tutorial-extensions.djangogirls.org/ko/manual\_pythonanywhere\_deploy/

static 자원 collect 하는 것과 DB migrate 하는 것은 생략함….