본인의 얼굴이 나오는 화면과 이름을 말한다.

로그인 => walab에 로그인

git clone : github 주소([https://github.com/jerry10004/2022OSS\_CRUD.git (Links to an external site.)](https://github.com/jerry10004/2022OSS_CRUD.git))를 clone 하여 로컬에 저장소를 복제 (ls로 파일 목록 확인 필수)  
⇒ walab 서버 ~/2022OSS/minitest/ 폴더에서 작업

코드 : mkdir minitest, cd minitest, git clone <https://github.com/jerry10004/2022OSS_CRUD.git>, ls

gitignore : .gitignore 파일 생성하여 object file과 실행파일(market)은 git에서 관리하지 않도록 추가.  
⇒ git add / commit

코드 : vi gitignore, [www.gitignore.io](http://www.gitignore.io)에서 복붙, git add gitignore, git commit

build & run : 각 라이브러리(manager.c, product.c)를 컴파일하여 목적파일로 생성하고, market.c를 빌드하여 market  실행파일을 만들어 보고 실행하여 1개 제품 추가/조회해 보기,  
⇒ git status

코드 : gcc -c manager.c product.c, gcc market.c manager.o product.o -o market, ./market, git status

Makefile 만들기 : 5번에서 연습해 본것을 바탕으로 Makefile 생성한 후, 실행  
⇒ make clean / make market  
⇒ 실행 완료되면 변경파일에 대한 이력 기록 : git add/commit

코드 : vim makefile, git add makefile, git commit

CC = gcc

TARGET = market.c

SRCS = product.c manager.c

OBJS = $(SRCS:.c=.o)

EXE1 = market

$(EXE1) : $(OBJS)

$(CC) $(TARGET) $(OBJS) -o $(EXE1)

%.o: %.c

$(CC) -c $(SRCS)

clean :

rm -f $(OBJS) $(EXE1

파일저장함수 구현 : saveData()함수 구현(manager.c에 있음)  
⇒ 구현/컴파일 테스트/테스트 완료되면 git add/commit

코드 : vim manager.c에 코드 추가, git add manager.c, git commit

FILE\* fp;

fp= fopen("product.txt","wt");

for (int i=0; i<count; i++) {

if(p[i].price==-1||p[i].weight==-1) continue;

fprintf(fp, "%s, %d %d\n", p[i].name, p[i].weight, p[i].price);

}

fclose(fp);

printf("저장됨!\n");

파일로드함수 구현 : loadData()함수 구현(manager.c에 있음)  
⇒ 구현/컴파일 테스트/테스트 완료되면 git add/commit

코드 : vim manager.c에 코드 추가, git add manager.c, git commit

int count=0;

FILE\*fp;

//파일 내용을 읽어와서 배열에 값 추가하기

fp=fopen("product.txt", "rt");

printf("=> 데이터 가져오는 중...\n");

if (fp==NULL) {

printf("=> 파일 없음 \n");

}

else {

for (int i=0;;i++){

fscanf(fp, "%[^,], %d %d", p[i].name, &p[i].weight, &p[i].price);

if (feof(fp)) break;

count++;

}

printf("=> %d 개의 주문 정보를 읽었습니다!\n", count);

printf("=> 로딩 성공!\n");

fclose(fp);

}

return count;

git remote repo 연결제거 : 현재 작업중인 로컬 저장소에 대해 github remote repository 연결 제거 작업  
⇒ git remote -v

코드 : git remote remove origin, git remote -v

github 저장소 생성 : github에 로그인하여 새로운 빈 저장소를 하나 생성 및 주소복사(github 새주소)

코드 : github 홈페이지에서 레파지토리 하나 추가 후 URL 복사

git remote repo 새연결 : walab 서버의 현재의 로컬 저장소에 github 새주소와의 remote repository 연결을 세팅

코드 : git remote add origin <URL>

git push : 현재 로컬 저장소에서 연결하고 있는 원격저장소(github 새주소)에 push

코드 : git push origin master

github 새로고침 : github 새주소에 들어가서 새로고침

코드 : github 홈페이지에서 새로고침