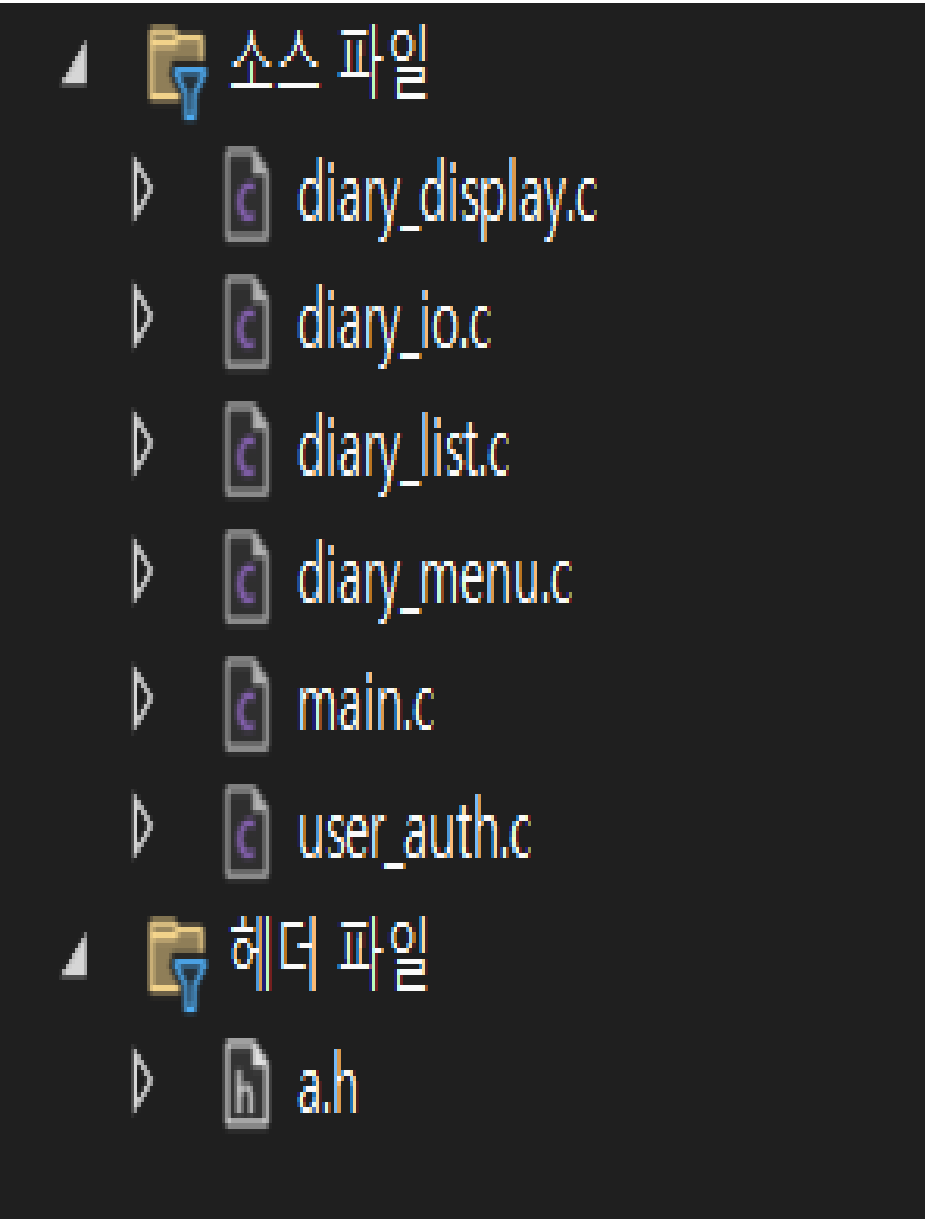
나만의 일기장

프로젝트 파일 구성



- diary_display.c : 화면에 보이는 여러 가지 테두리를 구성하는 소스파일

- diary_io.c : 일기장의 내용을 구성하는 입출력 소스파일 (io란 input/output의 약자)

- diary_list.c : 저장된 일기 목록을 보여주는 소스파일

- diary_menu.c : 일기장의 메뉴들을 구성하는 소스파일

- main.c : 일기장의 메인 소스파일

- user_auth.c : 일기장의 인증을 담당하는 소스파일 (auth란 Authentication의 약자)

- a.h : 일기장의 여러 가지 함수들을 정의하는 헤더파일 (모든 c파일이 include 하는 통합 헤더)

메인 파일 소개(main.c)

```c
int main() {
#ifdef _WIN32
    system("mkdir diary"); // 'diary' 폴더 생성 시도 (Windows 전용)
#endif

    MYSQL* db_conn = NULL; // MySQL 연결 포인터
    int page_width = 80;  // 화면 너비

    // 메인 프로그램 루프
    while (1) {
        // 데이터베이스 연결
        db_conn = init_db_connection();
        if (db_conn == NULL) {
            fprintf(stderr, "데이터베이스 연결에 실패했습니다. 프로그램을 종료합니다.Wn");
            return 1;
        }

        int logged_in = 0; // 로그인 상태 (0:로그아웃, 1:로그인, -1:프로그램 종료)

        // 로그인/회원가입/찾기 루프
        while (!logged_in) {
            display_auth_menu(); // 인증 메뉴 출력
            int auth_choice;
            scanf("%d", &auth_choice);
            getchar(); // 입력 버퍼 비우기

            // 인증 메뉴 선택 처리 함수 호출
            // 이 함수는 logged_in 상태를 변경하고, 프로그램 종료 여부(-1)를 반환할 수 있음
            int auth_result = handle_auth_choice(auth_choice, db_conn, &logged_in);

            if (auth_result == -1) { // 프로그램 종료 선택 시
                logged_in = -1; // 메인 루프도 종료하도록 logged_in 상태 변경
                break; // 인증 루프 종료
```

메인 파일 소개(main.c)

```
            system("pause");
            system("cls");
            // auth_result가 1이면 logged_in이 1로 설정되어 인증 루프 종료 (로그인 성공)
            // auth_result가 0이면 로그인 실패/회원가입 후이므로 인증 루프 계속
        }

        // 인증 루프에서 나왔을 때 (로그인 성공 또는 프로그램 종료)
        if (logged_in == -1) { // 프로그램 종료가 선택되었다면
            close_db_connection(db_conn); // DB 연결 닫기
            break; // 메인 프로그램 루프 종료
        }

        // --- 로그인 성공 후 일기장 메인 메뉴 ---
        int choice;
        while (logged_in == 1) { // 로그인 상태가 유지되는 동안
            display_main_menu(page_width); // 메인 일기장 메뉴 출력
            printf("선택: ");
            scanf("%d", &choice);
            getchar(); // 입력 버퍼 비우기

            // 일기장 메뉴 선택 처리 함수 호출
            // 이 함수는 logged_in 상태를 0(로그아웃)이나 -1(프로그램 종료)로 변경할 수 있음
            handle_menu_choice(choice, page_width, db_conn, &logged_in);
            system("pause");
            system("cls");
        }

        // 일기장 메뉴 루프에서 나왔을 때 (로그아웃 또는 프로그램 종료)
        close_db_connection(db_conn); // DB 연결 닫기
        // logged_in이 0이면 메인 루프의 처음으로 돌아가 다시 DB 연결 시도 및 인증 메뉴 표시
        // logged_in이 -1이면 메인 루프도 종료
        if (logged_in == -1) {
            break; // 메인 프로그램 루프 종료
        }
    }
```

주요 기능 및 화면 - 사용자 인증

```
MySQL 데이터베이스에 성공적으로 연결되었습니다.
'login' 테이블 확인/생성 완료.

**************************************
* 나만의 일기장 - 인증          *
**************************************
1. 로그인
2. 회원가입
3. 아이디/비밀번호 찾기
4. 프로그램 종료
선택: 
```

```c
// 인증 메뉴(로그인/회원가입/찾기)를 화면에 표시하는 함수
void display_auth_menu() {
    printf("₩n**************************************₩n");
    printf("* 나만의 일기장 - 인증          *₩n");
    printf("**************************************₩n");
    printf("1. 로그인₩n");
    printf("2. 회원가입₩n");
    printf("3. 아이디/비밀번호 찾기₩n");
    printf("4. 프로그램 종료₩n");
    printf("선택: ");
}
```

"프로그램 실행 시 가장 먼저 나타나는 사용자 인증 화면입니다. 사용자들은 여기서 로그인, 회원가입, 또는 계정 찾기 기능을 이용할 수 있습니다."

관련 코드 : diary_menu.c 의 diary_auth_menu()

주요 기능 및 화면 - 사용자 인증

"사용자가 로그인하는 코드입니다." 관련코드 : user_auth.c 의 login_user()

```
**************************************
* 나만의 일기장 - 인증              *
**************************************
1. 로그인
2. 회원가입
3. 아이디/비밀번호 찾기
4. 프로그램 종료
선택 : 1

--- 로그인 ---
ID를 입력하세요 : te
비밀번호를 입력하세요 : te
로그인 성공! 환영합니다, te님!
계속하려면 아무 키나 누르십시오 . . .
```

```
--- 로그인 ---
ID를 입력하세요 : qqq
비밀번호를 입력하세요 : qqqq
ID 또는 비밀번호가 잘못되었습니다.
계속하려면 아무 키나 누르십시오 . . .
```

```c
int success = 0; // 로그인 성공 여부를 저장할 변수 (초기값: 실패)
// 결과 행의 수가 0보다 크면 (즉, 일치하는 ID/비밀번호 조합이 존재하면)
if (mysql_num_rows(result) > 0) {
    printf("로그인 성공! 환영합니다, %s님!₩n", id);
    success = 1; // 성공
}
else { // 일치하는 조합이 없으면
    printf("ID 또는 비밀번호가 잘못되었습니다.₩n");
}

mysql_free_result(result); // 사용한 결과 셋 메모리 해제
return success; // 로그인 성공 여부 반환
```

```c
// 사용자 로그인 시도 함수
// conn: MySQL 연결 포인터
// 반환값: 1 (로그인 성공), 0 (로그인 실패)
int login_user(MYSQL* conn) {
    char id[21];     // 사용자 ID를 저장할 버퍼
    char paswwd[31]; // 비밀번호를 저장할 버퍼
    char query[200]; // SQL 쿼리 문자열을 저장할 버퍼
    MYSQL_RES* result; // SQL 쿼리 결과를 저장할 구조체 포인터

    printf("₩n--- 로그인 ---₩n");
    printf("ID를 입력하세요: ");
    if (fgets(id, sizeof(id), stdin) == NULL) return 0;
    CHOP(id);

    printf("비밀번호를 입력하세요: ");
    if (fgets(paswwd, sizeof(paswwd), stdin) == NULL) return 0;
    CHOP(paswwd);

    // 입력받은 ID와 비밀번호가 일치하는지 확인하는 SQL 쿼리 생성
    sprintf(query, "SELECT id FROM login WHERE id = '%s' AND paswwd = '%s'", id, paswwd);
    // SQL 쿼리 실행
    if (mysql_query(conn, query)) { // 쿼리 실행 실패 시
        fprintf(stderr, "SELECT query failed: %s₩n", mysql_error(conn));
        return 0; // 실패
    }

    // 쿼리 결과를 클라이언트로 가져옴
    result = mysql_store_result(conn);
    if (result == NULL) { // 결과 가져오기 실패 시
        fprintf(stderr, "mysql_store_result failed: %s₩n", mysql_error(conn));
        return 0; // 실패
    }
```

주요 기능 및 화면 - 사용자 인증

"사용자가 회원가입하는 코드입니다." 관련코드 : user_auth.c 의 register_user()

```
**************************************
* 나만의 일기장 - 인증              *
**************************************
1. 로그인
2. 회원가입
3. 아이디/비밀번호 찾기
4. 프로그램 종료
선택: 2

--- 회원가입 ---
사용할 ID를 입력하세요: hhh
사용할 비밀번호를 입력하세요: hhh
회원가입이 성공적으로 완료되었습니다.
계속하려면 아무 키나 누르십시오 . . .

**********************************
* 나만의 일기장 - 인증          *
**********************************
1. 로그인
2. 회원가입
3. 아이디/비밀번호 찾기
4. 프로그램 종료
선택: 2

--- 회원가입 ---
사용할 ID를 입력하세요: hhh
오류: 'hhh'는 이미 존재하는 ID입니다. 다른 ID를 사용해주세요.
계속하려면 아무 키나 누르십시오 . . .
```

```c
// 새로운 사용자를 데이터베이스에 등록(회원가입)하는 함수
// conn: MySQL 연결 포인터
// 반환값: 1 (회원가입 성공), 0 (회원가입 실패)
int register_user(MYSQL* conn) {
    char id[21];     // 사용자 ID를 저장할 버퍼 (VARCHAR(20) + 널문자)
    char paswwd[31]; // 비밀번호를 저장할 버퍼 (VARCHAR(30) + 널문자)
    char query[200]; // SQL 쿼리 문자열을 저장할 버퍼

    printf("₩n--- 회원가입 ---₩n");
    printf("사용할 ID를 입력하세요: ");
    // 사용자로부터 ID 입력 받기 (fgets는 개행 문자까지 읽어옴)
    if (fgets(id, sizeof(id), stdin) == NULL) return 0; // 입력 실패 시 0 반환
    CHOP(id); // 입력된 ID 문자열 끝의 개행 문자 제거

    // 입력받은 ID가 이미 존재하는지 확인
    if (check_id_exists(conn, id)) {
        printf("오류: '%s'는 이미 존재하는 ID입니다. 다른 ID를 사용해주세요.₩n", id);
        return 0; // 회원가입 실패
    }

    printf("사용할 비밀번호를 입력하세요: ");
    // 사용자로부터 비밀번호 입력 받기
    if (fgets(paswwd, sizeof(paswwd), stdin) == NULL) return 0;
    CHOP(paswwd); // 입력된 비밀번호 문자열 끝의 개행 문자 제거

// 사용자 정보를 'login' 테이블에 삽입하는 SQL 쿼리 생성
sprintf(query, "INSERT INTO login(id, paswwd) VALUES ('%s', '%s')", id, paswwd);
// SQL 쿼리 실행
if (mysql_query(conn, query)) { // 쿼리 실행 실패 시
    fprintf(stderr, "INSERT query failed: %s₩n", mysql_error(conn));
    return 0; // 실패
}

printf("회원가입이 성공적으로 완료되었습니다.₩n");
return 1; // 성공
```

주요 기능 및 화면 - 사용자 인증

"사용자의 아이디 비밀번호을 찾는코드입니다." 관련코드 user_auth.c 의 find_id_password()

```
선택: 3

--- 아이디/비밀번호 찾기 (모든 사용자 정보 출력) ---
--- (주의: 보안상 실제 서비스에서는 사용되지 않는 방식입니다.) ---

   ID           PASSWORD
------------------------------
skydream        aaabbb
홍길동          aaabbb
김광재          aaabbb
김수로          aaabbb
김광수          aaabbb
aaa             gewgew
huhuhu          jiijiji
a               a
gkgekg          qrkkqr
qwe             qwe
asd             asd
gagaq           zcxv
fafa            fafa
fa              fa
te              te
vvv             vvv
gg              ggg
ggg             ggg
hhh             hhh
------------------------------
모든 사용자 정보 출력이 완료되었습니다.
```

```c
// 아이디/비밀번호 찾기 함수 (디버깅 목적으로 모든 사용자 정보 출력)
// conn: MySQL 연결 포인터
void find_id_password(MYSQL* conn) {
    MYSQL_RES* result; // SQL 쿼리 결과를 저장할 구조체 포인터
    MYSQL_ROW row;     // 결과의 각 행(row)을 저장할 구조체 포인터

    printf("₩n--- 아이디/비밀번호 찾기 (모든 사용자 정보 출력) ---₩n");
    printf("--- (주의: 보안상 실제 서비스에서는 사용되지 않는 방식입니다.) ---₩n");

    // 'login' 테이블의 모든 ID와 비밀번호를 조회하는 SQL 쿼리 실행
    if (mysql_query(conn, "SELECT id, paswwd FROM login")) {
        fprintf(stderr, "조회 실패 !: %s₩n", mysql_error(conn));
        return; // 쿼리 실패 시 함수 종료
    }

    // 쿼리 결과를 클라이언트로 가져옴
    result = mysql_store_result(conn);
    if (result == NULL) { // 결과 가져오기 실패 시
        fprintf(stderr, "결과 가져오기 실패 !: %s₩n", mysql_error(conn));
        return; // 함수 종료
    }

printf("₩n   ID           PASSWORD₩n"); // 출력 헤더
printf("------------------------------₩n");
// 모든 결과 행을 하나씩 가져와서 출력
while ((row = mysql_fetch_row(result)) != NULL) {
    // row[0]은 첫 번째 컬럼 (ID), row[1]은 두 번째 컬럼 (비밀번호)
    // NULL 체크: 만약 DB에 값이 NULL로 저장되어 있다면 "NULL" 문자열 출력
    printf("%-15s %s₩n", row[0] ? row[0] : "NULL", row[1] ? row[1] : "NULL");
}
printf("------------------------------₩n");

mysql_free_result(result); // 사용한 결과 셋 메모리 해제 (필수!)
printf("모든 사용자 정보 출력이 완료되었습니다.₩n");
```

주요 기능 및 화면 – 로그인 후

"로그인 후 화면 구성입니다." display_menu_.c의 display_main_menu()

```
==============================================================================
##                          나만의 일기장 메뉴                                ##
==============================================================================
1. 새 일기 작성
2. 일기 목록 보기
3. 일기 보기
4. 로그아웃
5. 프로그램 종료
선택: |
```

```c
// 메인 메뉴를 화면에 표시하는 함수
void display_main_menu(int width) {
    draw_border(width, '=');
    printf("##");
    int menu_padding = (width - 4 - strlen(" 나만의 일기장 메뉴 ")) / 2;
    for (int i = 0; i < menu_padding; i++) printf(" ");
    printf(" 나만의 일기장 메뉴 ");
    for (int i = 0; i < width - 4 - menu_padding - strlen(" 나만의 일기장 메뉴 "); i++) printf(" ");
    printf("##₩n");
    draw_border(width, '=');

    printf("1. 새 일기 작성₩n");
    printf("2. 일기 목록 보기₩n");
    printf("3. 일기 보기₩n");
    printf("4. 로그아웃₩n");
    printf("5. 프로그램 종료₩n");
}
```

```c
// 지정된 너비와 문자로 테두리를 그리는 함수
void draw_border(int width, char character) {
    for (int i = 0; i < width; i++) {
        printf("%c", character); // 지정된 문자를 width만큼 반복 출력
    }
    printf("₩n"); // 줄 바꿈
}
```

"화면에 보이는 문자들을 반복문을 통해 만들어 주는 코드"

주요 기능 및 화면 – 로그인 후

" 일기를 작성하는 코드 입니다 . " 관련코드 : d i a r y _ i o . c의 w r i t e _ d i a r y ( )

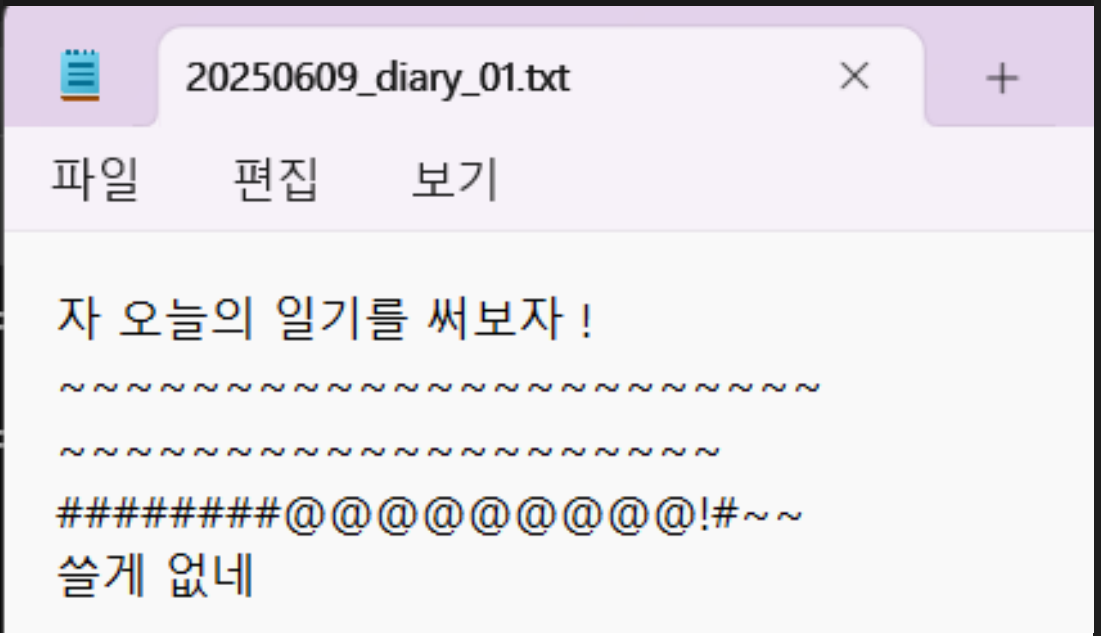
```
선택: 1

--- 새로운 일기 작성 ---
오늘 날짜: 2025년 06월 09일 (일기 1)
일기 내용을 입력하세요 (입력을 마치려면 새 줄에서 Ctrl+Z 후 Enter):
-------------------------
자 오늘의 일기를 써보자 !
~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
########@@@@@@@@@!#~~
쓸게 없네
^Z

일기가 'diary\20250609_diary_01.txt' 파일로 저장되었습니다.
```

```c
    printf("₩n--- 새로운 일기 작성 ---₩n");
    printf("오늘 날짜: %04d년 %02d월 %02d일 (일기 %d)₩n", year, month, day, sequence);
    printf("일기 내용을 입력하세요 (입력을 마치려면 새 줄에서 Ctrl+Z 후 Enter):₩n");
    printf("-------------------------₩n");

    char buffer[1024]; // 사용자 입력을 한 줄씩 읽을 임시 버퍼
    // EOF(End Of File) 문자 (Ctrl+Z)가 입력될 때까지 파일에 내용 쓰기
    while (fgets(buffer, sizeof(buffer), stdin) != NULL) {
        fputs(buffer, fp); // 읽은 내용을 파일에 쓰기
    }

    fclose(fp); // 파일 닫기
    printf("₩n일기가 '%s' 파일로 저장되었습니다.₩n", filename);
    getchar(); // 입력 버퍼 비우기 Ctrl+Z 입력 후 남은 개행 문자 처리)
}
```

```c
// 새로운 일기를 작성하고 파일에 저장하는 함수
void write_diary() {
    time_t t = time(NULL);       // 현재 시간 가져오기
    struct tm* tm = localtime(&t); // 현지 시간 구조체로 변환

    int year = tm->tm_year + 1900; // 년도 (1900 더해야 실제 년도)
    int month = tm->tm_mon + 1;    // 월 (0부터 시작하므로 1 더해야 함)
    int day = tm->tm_mday;         // 일

    // 현재 날짜의 다음 일기 순번 가져오기 (예: 1, 2, 3...)
    int sequence = get_next_diary_sequence(year, month, day);

    char filename[MAX_PATH]; // 저장할 파일명 버퍼
    // 파일명 형식: "diary/YYYYMMDD_diary_XX.txt" (예: diary/20250529_diary_01.txt)
    sprintf(filename, "diary₩₩%04d%02d%02d_diary_%02d.txt",
        year, month, day, sequence);

    FILE* fp = fopen(filename, "w"); // 파일 열기 (쓰기 모드, 파일 없으면 생성)
    if (fp == NULL) { // 파일 열기 실패 시
        printf("오류: 파일을 열 수 없습니다. (경로: %s)₩n", filename);
        printf("일기장 폴더(diary)가 있는지 확인해주세요.₩n");
        return;
    }
```

주요 기능 및 화면 – 로그인 후

" 일기를 작성 한 후 저장되는 폴더와 파일들 입니다 . "




```c
// 특정 날짜에 이미 작성된 일기의 다음 순번을 찾는 함수
// 예: 20250529_diary_01.txt, 20250529_diary_02.txt 가 있다면 3을 반환
int get_next_diary_sequence(int year, int month, int day) {
    char search_pattern[MAX_PATH]; // 검색할 파일 경로 패턴 버퍼
    // 'diary/YYYYMMDD_diary_*.txt' 형태의 패턴 생성 (예: diary/20250529_diary_*.txt)
    sprintf(search_pattern, "diary\\%04d%02d%02d_diary_*.txt", year, month, day);

    int max_sequence = 0; // 찾은 가장 큰 순번을 저장할 변수

    WIN32_FIND_DATAA findFileData; // 파일 검색 결과를 저장할 구조체 (Windows용)
    HANDLE hFind = INVALID_HANDLE_VALUE; // 파일 검색 핸들

    // 지정된 패턴으로 첫 번째 파일 검색
    hFind = FindFirstFileA(search_pattern, &findFileData);
    if (hFind == INVALID_HANDLE_VALUE) {
        return 1; // 해당 날짜의 일기가 없으면, 첫 번째 순번은 1
    }

// 파일이 하나라도 있다면, 다음 파일들을 계속 검색
do {
    // 현재 항목이 디렉토리가 아닌 일반 파일일 경우만 처리
    if (!(findFileData.dwFileAttributes & FILE_ATTRIBUTE_DIRECTORY)) {
        char* filename = findFileData.cFileName; // 찾은 파일 이름
        char* suffix_ptr = strstr(filename, "_diary_"); // 파일명에서 "_diary_" 문자열 찾기
        if (suffix_ptr != NULL) {
            // "_diary_" 바로 뒤에 오는 숫자(순번)를 정수로 변환
            int current_seq = atoi(suffix_ptr + strlen("_diary_"));
            // 현재 순번이 최대 순번보다 크면 갱신
            if (current_seq > max_sequence) {
                max_sequence = current_seq;
            }
        }
    }
} while (FindNextFileA(hFind, &findFileData) != 0); // 다음 파일이 없을 때까지 반복

FindClose(hFind); // 파일 검색 핸들 닫기 (메모리 누수 방지)

return max_sequence + 1; // 찾은 최대 순번에 1을 더해 다음 순번 반환
```

주요 기능 및 화면 – 로그인 후

" 저장된 일기 목록을 보여주는 코드 입니다 . "

```
선택: 2

--- 저장된 일기 목록 ---
20250609 - 일기 1
20250609 - 일기 2
20250609 - 일기 3
------------------------
계속하려면 아무 키나 누르십시오 . . .
```

관련코드 : diary_list.c의 list_diares()

```c
// 저장된 일기 파일들의 목록을 읽어와 화면에 표시하는 함수
void list_diaries() {
    printf("\n--- 저장된 일기 목록 ---\n");

    WIN32_FIND_DATAA findFileData; // 파일 검색 결과를 저장할 구조체
    HANDLE hFind = INVALID_HANDLE_VALUE; // 파일 검색 핸들
    char searchPath[MAX_PATH]; // 검색할 경로 버퍼 (Windows용 최대 경로 길이)

    // 'diary' 폴더 안의 모든 '.txt' 파일을 검색할 패턴 생성
    sprintf(searchPath, "diary\\*.txt");

    // 지정된 패턴으로 첫 번째 파일 검색 시작
    hFind = FindFirstFileA(searchPath, &findFileData);
    if (hFind == INVALID_HANDLE_VALUE) { // 파일 검색 실패 (폴더 없거나 파일 없음)
        printf("오류: 'diary' 폴더를 찾을 수 없거나 파일이 없습니다.\n");
        printf("------------------------\n");
        return;
    }

    int count = 0; // 찾은 일기 파일의 개수
    do {
        // 현재 항목이 디렉토리가 아닌 일반 파일일 경우만 처리
        if (!(findFileData.dwFileAttributes & FILE_ATTRIBUTE_DIRECTORY)) {
            char* filename = findFileData.cFileName; // 찾은 파일 이름
            // 파일명이 "_diary_"를 포함하고 ".txt"로 끝나는지 확인
            if (strstr(filename, "_diary_") != NULL && strstr(filename, ".txt") != NULL) {
                char date_str[9]; // 날짜 부분 (YYYYMMDD)을 저장할 버퍼
                int sequence;     // 순번을 저장할 변수
                // 파일명에서 날짜와 순번을 파싱 (예: "20250527_diary_01.txt" -> "20250527", 1)
                sscanf(filename, "%8s_diary_%d.txt", date_str, &sequence);

                // 화면에 "날짜 - 일기 순번" 형식으로 출력
                printf("%s - 일기 %d\n", date_str, sequence);
                count++;
            }
        }
    } while (FindNextFileA(hFind, &findFileData) != 0); // 다음 파일이 없을 때까지 반복

    FindClose(hFind); // 파일 검색 핸들 닫기

    if (count == 0) { // 찾은 일기가 없으면
        printf("저장된 일기가 없습니다.\n");
    }
    printf("------------------------\n");
}
```

주요 기능 및 화면 – 로그인 후

" 저장된 일기를 보여주는 코드 입니다 . " 관련코드 : d i a r y _ i o . C의 v i e w _ d i a r y ( )

```
선택: 3

--- 일기 보기 ---
보고 싶은 일기 날짜와 순번을 입력하세요 (예: 20250527 1): 20250609 1

********************************************************************************
********************************************************************************
##                              20250609 일기 1                                 ##
********************************************************************************
********************************************************************************
*                                                                              *
*                              20250609 일기 1                                 *
*                                                                              *
* 자 오늘의 일기를 써보자 !                                                   *
* ~~~~~~~~~~~~~~~~~~~~~~~~                                                     *
* ~~~~~~~~~~~~~~~~~~~~                                                         *
* ########@@@@@@@@!#~~                                                        *
* 쓸게 없네                                                                   *
*                                                                              *

********************************************************************************

일기 보기가 완료되었습니다.
계속하려면 아무 키나 누르십시오 . . .
```

```c
// 특정 날짜/순번의 일기 내용을 읽어와 화면에 표시하는 함수
void view_diary(int page_width) {
    char search_date_str[9]; // 검색할 날짜 문자열 (YYYYMMDD)
    int search_sequence;     // 검색할 일기 순번
    char filename[MAX_PATH]; // 검색할 파일명 버퍼
    char line_buffer[1024];  // 파일에서 한 줄씩 읽을 임시 버퍼
    char content_buffer[10000]; // 읽어온 일기 내용 전체를 저장할 버퍼 (넉넉하게)
    content_buffer[0] = '₩0'; // 버퍼 초기화 (빈 문자열로)

    printf("₩n--- 일기 보기 ---₩n");
    printf("보고 싶은 일기 날짜와 순번을 입력하세요 (예: 20250527 1): ");
    scanf("%8s %d", search_date_str, &search_sequence); // 날짜와 순번을 함께 입력받음
    getchar(); // 입력 버퍼 비우기

    // 검색할 파일명 생성 (예: diary/20250527_diary_01.txt)
    sprintf(filename, "diary₩₩%s_diary_%02d.txt", search_date_str, search_sequence);

    FILE* fp = fopen(filename, "r"); // 파일 열기 (읽기 모드)
    if (fp == NULL) { // 파일 열기 실패 (파일이 없거나 읽을 수 없음)
        printf("오류: '%s' 일기를 찾을 수 없습니다.₩n", filename);
        printf("날짜와 순번 형식을 확인하거나 '일기 목록 보기'를 통해 확인해주세요.₩n");
        return;
    }
```

주요 기능 및 화면 – 로그인 후

" 저장된 일기를 보여주는 코드 입니다 . " 관련코드 : diary_io.C의 view_diary()

```
선택: 3

--- 일기 보기 ---
보고 싶은 일기 날짜와 순번을 입력하세요 (예: 20250527 1): 20250609 1

********************************************************************************
********************************************************************************
##                          20250609 일기 1                                   ##
********************************************************************************
********************************************************************************
*                                                                              *
*                          20250609 일기 1                                     *
*                                                                              *
* 자 오늘의 일기를 써보자 !                                                   *
* ~~~~~~~~~~~~~~~~~~~~~~~~~                                                    *
* ~~~~~~~~~~~~~~~~~~~                                                          *
* ########@@@@@@@@!#~~                                                        *
* 쓸게 없네                                                                   *
*                                                                              *

********************************************************************************

일기 보기가 완료되었습니다.
계속하려면 아무 키나 누르십시오 . . .
```

```c
printf("₩n");
draw_border(page_width, '*'); // 상단 '*' 테두리 그리기
printf("********************************************************************************₩n"); // 추가 디자인 라인
printf("##");
char display_title[50]; // 화면에 표시할 제목 버퍼
// 제목 생성: "YYYYMMDD 일기 X" (예: 20250527 일기 1)
sprintf(display_title, "%s 일기 %d", search_date_str, search_sequence);
int title_len = strlen(display_title);
int title_padding = (page_width - 4 - title_len) / 2; // 제목 중앙 정렬을 위한 공백 계산
for (int i = 0; i < title_padding; i++) printf(" ");
printf(" %s ", display_title); // 제목 출력
for (int i = 0; i < page_width - 4 - title_padding - title_len; i++) printf(" ");
printf("##₩n");
printf("********************************************************************************₩n"); // 추가 디자인 라인
draw_border(page_width, '*'); // 또 다른 상단 '*' 테두리 그리기

// 파일 내용을 한 줄씩 읽어 content_buffer에 모두 저장
while (fgets(line_buffer, sizeof(line_buffer), fp) != NULL) {
    strcat(content_buffer, line_buffer); // 읽은 줄을 전체 내용 버퍼에 추가
}

fclose(fp); // 파일 닫기

// 저장된 전체 일기 내용을 화면에 예쁘게 출력
print_diary_block(display_title, content_buffer, page_width);

printf("₩n");
draw_border(page_width, '*'); // 하단 '*' 테두리 그리기
printf("₩n일기 보기가 완료되었습니다.₩n");
```

주요 기능 및 화면 – 로그인 후

"저장된 일기의 테두리를 형성하는 코드 입니다." 관련코드 : diary_display.c의 print_diary_block()

```
보고 싶은 일기 날짜와 순번을 입력하세요 (예: 20250527 1): 20250609 3

**********************************************************************
**********************************************************************
##                          20250609 일기 3                         ##
**********************************************************************
**********************************************************************
*                                                                    *
*                          20250609 일기 3                           *
*                                                                    *
* qegqegq                                                           *
* qegeqgqeg                                                         *
* qegqeg                                                            *
* qegqeg                                                            *
* qegqeg                                                           *
* qegqegqegqeg                                                      *
* qegqegqegsg                                                       *
* qegeqgqeg                                                         *
* qegqgeqgq                                                         *
* qegqegqeg                                                         *
* qegqeg                                                            *
* qegqegag                                                          *
* qegqegqegasge                                                     *
* qegqegqegqeg                                                      *
*                                                                    *

**********************************************************************

일기 보기가 완료되었습니다.
계속하려면 아무 키나 누르십시오 . . .
```

```c
// 일기 내용을 예쁜 박스 형태로 화면에 출력하는 함수
void print_diary_block(const char* title, const char* content, int width) {
    int content_len = strlen(content); // 일기 내용의 길이
    int title_len = strlen(title);     // 제목의 길이

    // 제목 중앙 정렬을 위한 양쪽 공백 계산
    // width - 4는 양쪽 '*' 문자와 그 옆의 공백을 제외한 실제 내용 영역
    int title_padding = (width - 4 - title_len) / 2;

    // 상단 빈 줄 출력
    printf("*");
    for (int i = 0; i < width - 2; i++) {
        printf(" ");
    }
    printf("*₩n");

    // 제목 줄 출력 (중앙 정렬)
    printf("*");
    for (int i = 0; i < title_padding; i++) printf(" "); // 제목 앞 공백
    printf(" %s ", title); // 제목 출력
    // 제목 뒤 공백 (전체 너비에서 앞 공백, 제목, 양옆 한 칸 공백 제외)
    for (int i = 0; i < width - 4 - title_padding - title_len; i++) printf(" ");
    printf("*₩n");

    // 제목 아래 빈 줄 출력
    printf("*");
    for (int i = 0; i < width - 2; i++) {
        printf(" ");
    }
    printf("*₩n");
```

주요 기능 및 화면 – 로그인 후

"저장된 일기의 테두리를 형성하는 코드 입니다." 관련코드 : diary_display.c의 print_diary_block()

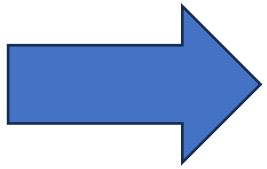
```
보고 싶은 일기 날짜와 순번을 입력하세요 (예: 20250527 1): 20250609 3

********************************************************************************
********************************************************************************
##                          20250609 일기 3                                   ##
********************************************************************************
********************************************************************************
*                                                                              *
*                          20250609 일기 3                                     *
*                                                                              *
* qegqegq                                                                      *
* qegeqgqeg                                                                    *
* qegqeg                                                                       *
* qegqeg                                                                       *
* qegqeg                                                                       *
* qegqegqegqeg                                                                 *
* qegqegqegsg                                                                  *
* qegeqgqeg                                                                    *
* qegqgeqgq                                                                    *
* qegqegqeg                                                                    *
* qegqeg                                                                       *
* qegqegag                                                                     *
* qegqegqegasge                                                                *
* qegqegqegqeg                                                                 *
*                                                                              *

********************************************************************************

일기 보기가 완료되었습니다.
계속하려면 아무 키나 누르십시오 . . .
```

```c
// 내용 라인 출력 (줄 바꿈 처리 및 좌측 정렬)
// 동적 메모리 할당: 내용 한 줄을 담을 버퍼 (width - 4는 내용 실제 영역)
char* buffer = (char*)malloc(sizeof(char) * (width - 4 + 1));
if (buffer == NULL) { // 메모리 할당 실패 시 오류 메시지 출력 후 종료
    perror("메모리 할당 실패");
    return;
}

int current_pos = 0; // 현재 읽고 있는 내용의 위치
while (current_pos < content_len) {
    printf("* "); // 왼쪽 테두리 '*'와 공백
    int char_count = 0; // 버퍼에 담은 문자 수
    // 한 줄에 출력할 수 있는 최대 문자 수 (width - 4)만큼 내용을 버퍼에 담기
    for (int i = 0; i < width - 4 && current_pos < content_len; i++) {
        // 내용 중 줄 바꿈 문자('Wn')를 만나면 강제로 줄 바꿈
        if (content[current_pos] == 'Wn') {
            buffer[char_count] = 'W0'; // 버퍼 끝에 널 문자 추가
            current_pos++; // 'Wn' 문자 건너뛰기
            break; // 현재 줄 처리 완료
        }
        buffer[i] = content[current_pos]; // 문자를 버퍼에 복사
        current_pos++; // 다음 문자로 이동
        char_count++;  // 문자 수 증가
    }
    buffer[char_count] = 'W0'; // 버퍼 끝에 널 문자 추가 (문자열 종료)

    // 버퍼 내용을 화면에 출력 (width - 4만큼 좌측 정렬)
    printf("%-*s", width - 4, buffer);
    printf(" *Wn"); // 오른쪽 테두리 '*'와 줄 바꿈
}

free(buffer); // 동적으로 할당한 메모리 해제 (매우 중요!)
```

주요 기능 및 화면 – 로그인 후

"마지막으로 로그아웃 기능의 코드입니다." 관련 코드 : diary_menu.c의 handle_menu_choice()

```
선택: 4

로그아웃되었습니다. 다시 로그인/회원가입 화면으로 돌아갑니다.
계속하려면 아무 키나 누르십시오 . . .
```



```
MySQL 연결이 해제되었습니다.
MySQL 데이터베이스에 성공적으로 연결되었습니다
'login' 테이블 확인/생성 완료.

****************************************
* 나만의 일기장 - 인증                 *
****************************************
1. 로그인
2. 회원가입
3. 아이디/비밀번호 찾기
4. 프로그램 종료
선택:
```

```c
void handle_menu_choice(int choice, int page_width, MYSQL* db_conn, int* logged_in_status) {
    switch (choice) {
    case 1:
        write_diary();
        break;
    case 2:
        list_diaries();
        break;
    case 3:
        view_diary(page_width);
        break;
    case 4: // 로그아웃
        printf("\n로그아웃되었습니다. 다시 로그인/회원가입 화면으로 돌아갑니다.\n");
        *logged_in_status = 0; // 로그인 상태를 0(로그아웃)으로 변경
        break;
    case 5: // 프로그램 종료
        printf("\n일기장을 종료합니다. 안녕히 계세요!\n");
        *logged_in_status = -1; // 프로그램 종료를 의미하는 특수 값
        break;
    default:
        printf("잘못된 선택입니다. 다시 입력해주세요.\n");
        break;
    }
}
```

감사합니다.