

C프로그래밍및실습

# 생성형 AI를 이용한 메세지 전송 시스템

진척 보고서 #1

제출일자:23.11.26.

제출자명:김건형

제출자학번:214249

## 1. 프로젝트 목표

### 1) 배경 및 필요성

운전을 하거나 급한 용무가 있을 때에는 메시지를 길게 보내는 것이 불편하다. 이런 경우에는 키워드만 간단하게 작성하여 원하는 메시지를 전달하는 기능이 필요하다 느껴본 적이 있을 것이다. 이런 기능을 구현하기 위해 생성형 AI를 이용하여 메시지 내용을 생성하고 저장해보기로 한다

### 2) 프로젝트 목표

생성형 AI API를 이용하여 사용자가 입력한 키워드를 가지고 메시지를 생성한다. 생성된 메시지를 배열에 저장하고 읽을 수 있도록 한다. 또한 특정 키를 입력받아 실제 메신저로 메시지를 전송하는 기능을 제공한다.

### 3) 차별점

기존에는 긴 메시지를 작성해야 할 때 직접 작성했다면 이 프로젝트를 통해서 메시지를 키워드만 가지고도 생성하고 전송할 수 있다. 생성형 AI의 특성상 원하지 않는 메시지가 전달될 가능성도 존재하기 때문에, 해당 메시지의 마지막에 본 메시지가 AI를 통해 생성되었다는 문구를 추가한다.

## 2. 기능 계획

### 1) 메시지 생성 기능

- 사용자의 키워드를 입력받아 메시지를 생성한다.

- (1) 사용자의 키워드 입력받기

- 사용자에게 n개의 키워드를 입력받고 배열로 저장한다.

- (2) 생성형 AI API 연결하기

- 생성형 AI API를 연결하여 키워드를 전송하고 생성형 AI API 기능에 포함된 말투 설정 (role설정을 통해서 메시지 수신자와의 관계를 매개변수로 넘겨줄 수 있음)을 통해 사용자가 원하는 길이의 메시지를 생성한다.

#예시

수신자: 아빠

입력 키워드: 곧, 도착

생성형 AI API 매개변수: [입력 키워드를 가지고 최대 N개의 단어로 문장 생성해줘, role: 아들]

출력 메시지: 아빠 저 곧 도착해요

### (3) 생성된 메시지 저장하기

- 사용자의 키워드를 통해 생성된 메시지를 배열로 저장한다.
- 저장하는 메시지의 마지막에 AI를 통해 생성된 메시지라는 문구를 추가한다.

## 2) 메시지 전송 기능

- 특정 키를 함께 입력하여 프로그램을 실행하면 메시지를 실제로 전송할 수 있도록한다.

### (1) 메신저 API 연결

- 메신저 API를 연결하여 사용자가 메시지를 실제로 전송할 수 있도록 한다.

## 3. 진척사항

### 1) 기능 구현

#### (1) 사용자의 키워드 입력받기

- 입력: 키워드 입력 선택(번호), 키워드, 키워드 입력 종료 문자.

출력: 사용자가 입력한 키워드

- 사용자가 키워드를 입력하고자하면 특정 번호를 입력하여 키워드를 입력할 수 있도록

하고, 키워드 입력이 종료되었음을 알리는 문자(!)를 통해서 사용자가 키워드 입력을 마치면 입력한 키워드를 출력하여 확인시켜줍니다.

- 키워드 입력 선택을 구분하는 과정에서 조건문, 키워드 입력을 받는 부분에서 반복문, 배열을 선언하는 부분에서 동적배열로 할당하였기 때문에 포인터의 개념이 사용됨.
- 코드 스크린샷

```
#include <stdio.h>
#include <stdlib.h>

#define MAX_KEYWORD_SIZE 128
#define MAX_KEYWORD_COUNT 20
#define MAX_MESSEGE_SIZE 256
#define MAX_MESSEGE_COUNT 256

// 키워드를 입력받고, 입력받은 키워드의 개수를 반환한다.
int input_keywords(char **keywords) {
    // 몇 개의 키워드가 있는지 저장할 변수
    int keyword_count = 0;
    printf(
        "최대 %d 글자의 키워드를 입력하세요. (입력을 마치려면 !를 입력하세요)\n",
        MAX_KEYWORD_SIZE);
    for (int i = 0; i < MAX_KEYWORD_COUNT; i++) {
        printf("%d번째 키워드: ", i + 1);
        scanf("%s", keywords[i]);

        if (keywords[i][0] == '!') {
            keyword_count = i;
            printf("키워드 입력이 끝났습니다.\n\n");
            break;
        }
    }

    printf("입력하신 키워드는 총 %d개 입니다.\n", keyword_count);
    for (int i = 0; i < keyword_count; i++) {
        printf("%s", keywords[i]);
        if (i < keyword_count - 1) {
            printf(", ");
        } else {
            printf("\n");
        }
    }

    return keyword_count;
}

int main() {
    // 메시지를 저장할 배열
    // char messages[MAX_MESSEGE_COUNT][MAX_MESSEGE_SIZE];
    // 동적할당으로 변경
    char **messages = (char **)malloc(MAX_MESSEGE_COUNT * (int)sizeof(char *));
    for (int i = 0; i < MAX_MESSEGE_COUNT; i++) {
        messages[i] = (char *)malloc(MAX_MESSEGE_SIZE * (int)sizeof(char *));
    }

    // 키워드를 저장할 배열
    // char keywords[MAX_MESSEGE_COUNT][MAX_KEYWORD_SIZE];
    // 동적할당으로 변경
    char ***keywords = (char ***)malloc(MAX_MESSEGE_COUNT * (int)sizeof(char **));
    for (int i = 0; i < MAX_MESSEGE_COUNT; i++) {
        keywords[i] = (char **)malloc(MAX_KEYWORD_COUNT * (int)sizeof(char *));
        for (int j = 0; j < MAX_KEYWORD_COUNT; j++) {
            keywords[i][j] = (char *)malloc(MAX_KEYWORD_SIZE * (int)sizeof(char));
        }
    }

    // 메시지의 개수, 키워드의 개수를 저장할 변수
    int messege_count = 0;
    int keyword_count = 0;

    // 선택한 메뉴를 저장할 변수
    int menu = 0;

    printf("*****자동 메시지 생성기*****\n\n");

    while (1) {
        printf("-----메뉴를 선택하세요-----\n");
        printf("1. 키워드로 메시지 생성하기\n");
        printf("2. 메시지 출력하기\n");
        printf("3. 메시지 재생성하기\n");
        printf("4. 메시지 삭제하기\n");
        printf("5. 종료하기\n");
        scanf("%d", &menu);
        // break;

        if (menu == 1) {
            // 키워드를 입력받고 개수를 변수에 저장한다.
            keyword_count = input_keywords(keywords[messege_count]);

            // TODO. 메시지 생성하는 기능 추가

        } else if (menu == 2) {
            // TODO. 메시지 출력하는 기능 추가

        } else if (menu == 3) {
            // TODO. 메시지 재생성하는 기능 추가

        } else if (menu == 4) {
            // TODO. 메시지 삭제하는 기능 추가

        } else if (menu == 5) {
            printf("프로그램을 종료합니다.\n");
            break;
        }
    }
}
```

## 2) 테스트 결과

### (1) 테스트한 기능 이름

- 세부기능 1. 사용자의 키워드 입력받기

```

*****자동 메세지 생성기*****

-----메뉴를 선택하세요-----
1. 키워드로 메세지 생성하기
2. 메세지 출력하기
3. 메세지 재생성하기
4. 메세지 삭제하기
5. 종료하기
1
최대 128 글자의 키워드를 입력하세요. (입력을 마치려면 !를 입력하세요)
1번째 키워드 : 곧
2번째 키워드 : 도착
3번째 키워드 : 광주역
4번째 키워드 : !
키워드 입력이 끝났습니다.

입력하신 키워드는 총 3개 입니다.
곧 , 도착 , 광주역

```

#### 4. 계획 대비 변경 사항

해당사항 없음

#### 5. 프로젝트 일정

업무		11/3	11/10	11/17	11/30	12/5	12/15	12/21
제안서 작성		완료						
기능1	세부기능1		완료					
	세부기능2			----->				
	세부기능3				----->			
기능2	세부기능1					----->		
유지보수 및 테스트							----->	