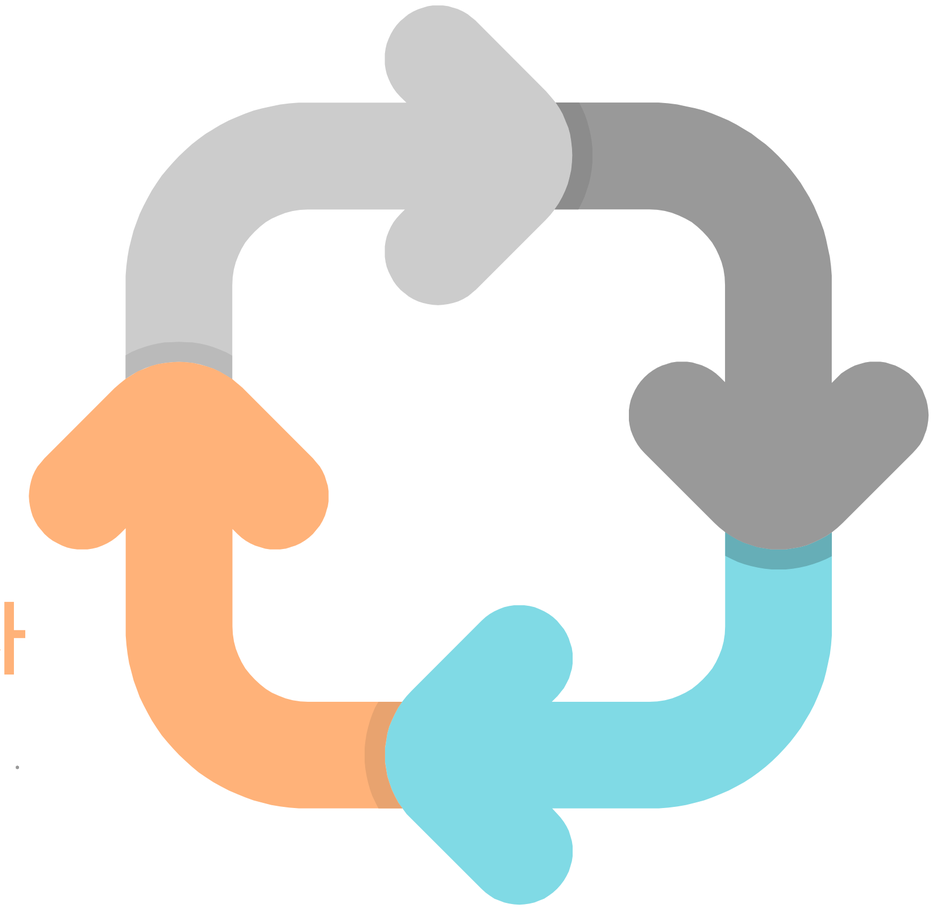
고품질 직업교육의 실현
경북산업직업전문학교

요리 레시피

김동현,조민석

목차

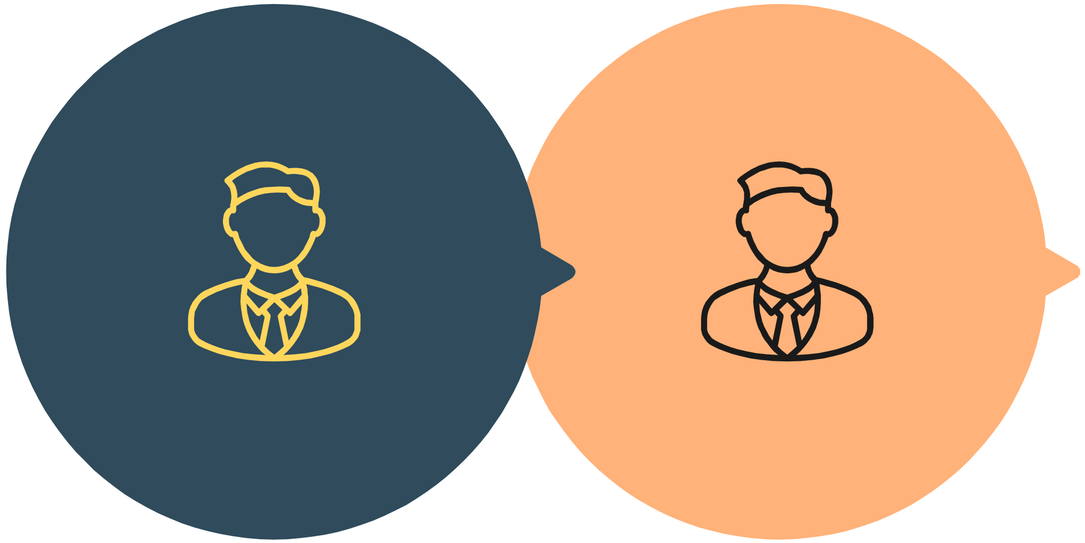


개요

요리 레시피

현대 자취생들은 배달을 시켜 먹기 마련인데 요리레시피를
알려주는 챗 봇 으로 가끔이라도 요리를 만들어 먹게 하기 위함

팀원 역활



01

김동현

. 기본 코드 토대 제작 및 기능구현

. 카카오 챗봇 시나리오 제작

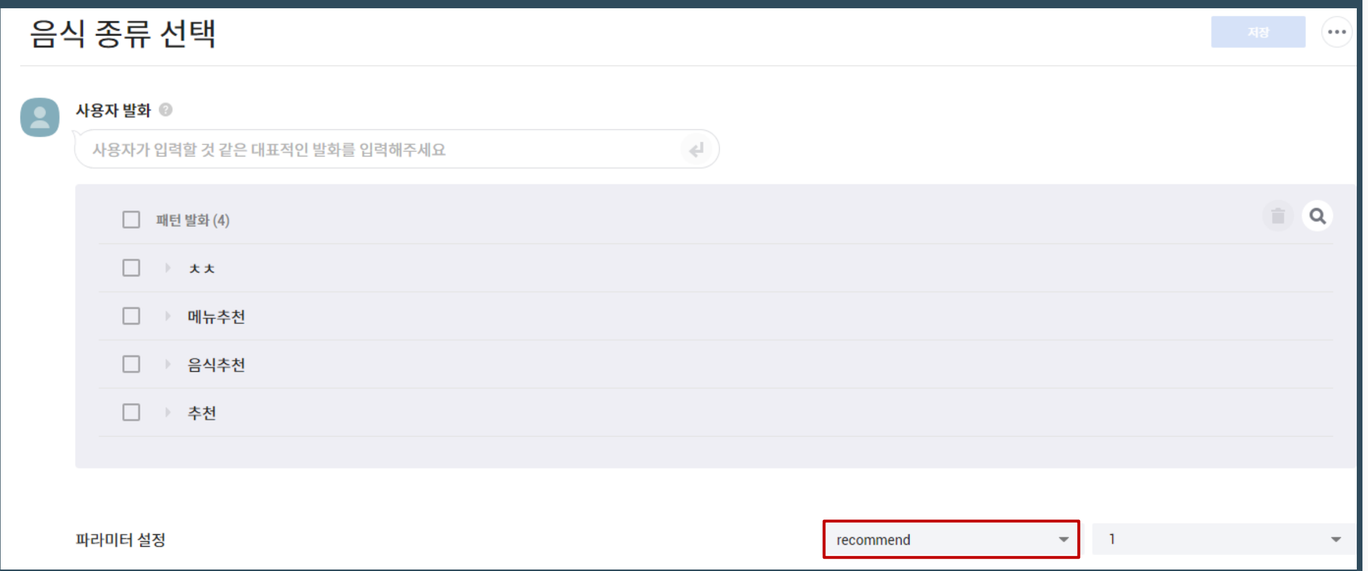
02

조민석

. Ppt 제작 및 발표

. 코드 수정 및 시나리오 수정

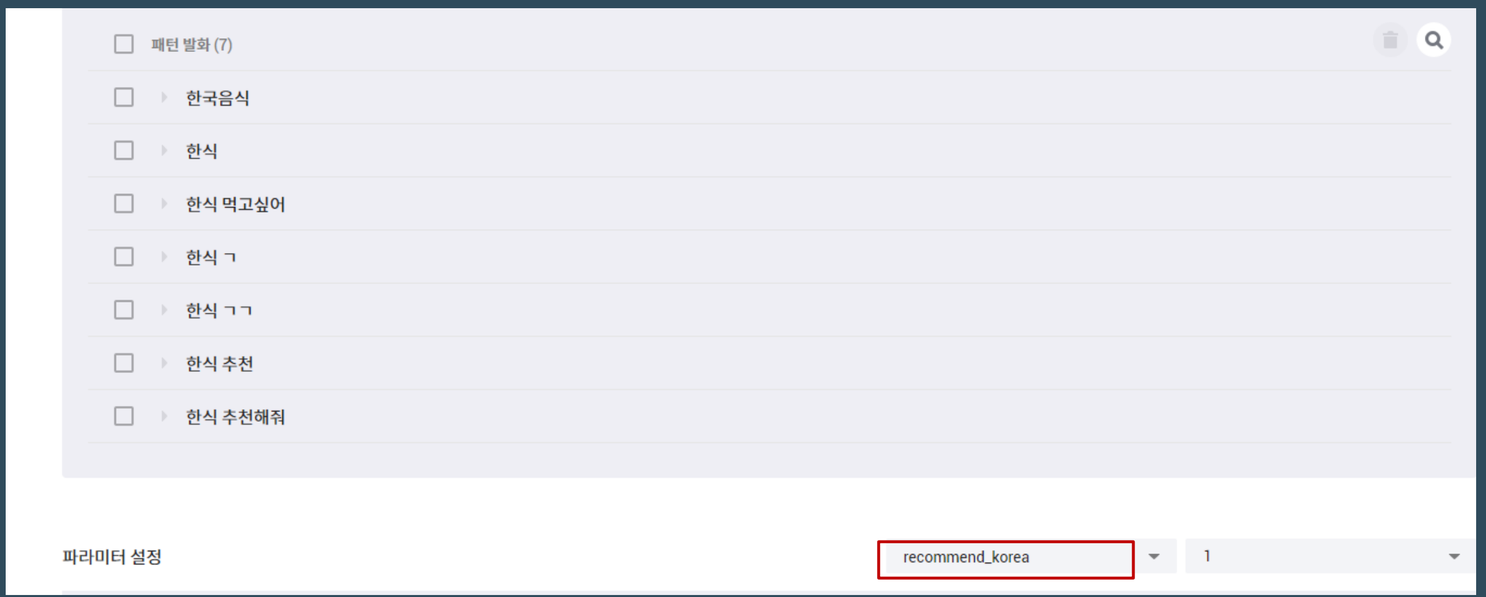
코드



메뉴 추천 입력 시 recommend 스킬이 발화 5개의 선택지를 라벨로 출력

```python
@application.route("/recommend", methods=["POST"])
def handle_request1():

    if request.path == "/recommend":
        message_text = "음식 분류를 선택해주세요."
        quick_replies = [
            {"messageText": "한식", "action": "message", "label": "한식"},
            {"messageText": "중식", "action": "message", "label": "중식"},
            {"messageText": "일식", "action": "message", "label": "일식"},
            {"messageText": "양식", "action": "message", "label": "양식"},
            {"messageText": "상관없음", "action": "message", "label": "상관없음"}
        ]

        res = {
            "version": "2.0",
            "template": {
                "outputs": [
                    {
                        "simpleText": {
                            "text": message_text
                        }
                    }
                ],
                "quickReplies": quick_replies
            }
        }
        return jsonify(res)
    else:
        return jsonify({"error": "Invalid input for recommendation"})
```

코드



한식을 선택했다면 설정한 패턴의 한식이 발화하여 recommend_korea 스킬

```python
@application.route("/recommend_korea", methods=["POST"])
def handle_request2():
    ran = random.randint(0, 4)

    youtubeUrl = youtube.korea(ran)
    blogUrl = blog.korea(ran)
    getMenu = menu.korea(ran)
    res = {
        "version": "2.0",
        "template": {
            "outputs": [
                {
                    "basicCard": {
                        "title": f"{getMenu}을(를) 추천드립니다.",
                        "buttons": [
                            {
                                "action": "webLink",
                                "label": "유튜브로 레시피 검색",
                                "webLinkUrl": youtubeUrl
                            },
                            {
                                "action": "webLink",
                                "label": "블로그로 레시피 검색",
                                "webLinkUrl": blogUrl
                            }
                        ]
                    }
                }
            ]
        }
    }
    return jsonify(res)
```

코드

중식을 선택했다면 설정한 패턴의 중식이 발화하여 recommend_china 스킬

```python
@application.route("/recommend_china", methods=["POST"])
def handle_request3():
    ran = random.randint(0, 4)

    youtubeUrl = youtube.china(ran)
    blogUrl = blog.china(ran)
    getMenu = menu.china(ran)
    res = {
        "version": "2.0",
        "template": {
            "outputs": [
                {
                    "basicCard": {
                        "title": f"{getMenu}을(를) 추천드립니다.",
                        "buttons": [
                            {
                                "action": "webLink",
                                "label": "유튜브로 레시피 검색",
                                "webLinkUrl": youtubeUrl
                            },
                            {
                                "action": "webLink",
                                "label": "블로그로 레시피 검색",
                                "webLinkUrl": blogUrl
                            }
                        ]
                    }
                }
            ]
        }
    }
    return jsonify(res)
```

코드

```
menu = menu.MENU()
youtube = youtube.RANTUBE()
blog = blog.RANBLOG()
```

```
class MENU:
    def __init__(self):
        return None

    def korea(self, ran):
        kFood = [
            '김치찌개',
            '된장찌개',
            '미역국',
            '떡볶이',
            '김치볶음밥'
        ]
        return kFood[ran]
```

```
class RANTUBE:
    def __init__(self):
        return None

    def korea(self, ran):
        kFood = [
            # 김치찌개
            'https://www.youtube.com/watch?v=qWbHSOplcvY',
            # 된장찌개
            'https://www.youtube.com/watch?v=ffuakdFmuh4',
            # 미역국
            'https://www.youtube.com/watch?v=xsTFsunt6-8',
            # 떡볶이
            'https://www.youtube.com/watch?v=t4Es8mwdYlE',
            # 김치볶음밥
            'https://www.youtube.com/watch?v=eIo2BaE6LxI'
        ]
        return kFood[ran]
```

```
class RANBLOG:
    def __init__(self):
        return None

    def korea(self, ran):
        kFood = [
            # 김치찌개
            'https://blog.naver.com/mj_hudadak/223494136932',
            # 된장찌개
            'https://blog.naver.com/peace8012/223495500311',
            # 미역국
            'https://blog.naver.com/peace8012/223491181911',
            # 떡볶이
            'https://blog.naver.com/mj_hudadak/223453534021',
            # 김치볶음밥
            'https://blog.naver.com/peace8012/223471907045'
        ]
        return kFood[ran]
```

랜덤 변수를 매개 변수로 보내주어 리스트의 값 중 하나를 랜덤 하게 가져 감

코드


```python
@application.route("/search", methods=["POST"])
def searchFood():
    global food
    req = request.get_json()
    food = req["action"]["detailParams"]["음식명"]["value"]  # 대표 엔티티로 지정
    res = {
        "version": "2.0",
        "template": {
            "outputs": [
                {
                    "basicCard": {
                        "title": f"{food} 레시피를 찾으세요?",
                        "buttons": [
                            {
                                "action": "message",
                                "label": "유튜브로 레시피 검색",
                                "messageText": "유튜브로 검색"
                            },
                            {
                                "action": "message",
                                "label": "블로그로 레시피 검색",
                                "messageText": "블로그로 검색"
                            }
                        ]
                    }
                }
            ]
        }
    }
    return jsonify(res)
```

카카오 챗봇에 엔티티를 설정하여 음식 이름을 입력하면 유튜브로 검색 버튼과 블로그로 검색 버튼이 출력

코드

```python
@application.route("/search_youtube", methods=["POST"])
def searchYoutube():
    global food

    if food is None:  # 채팅에 바로 '유튜브로 검색'을 입력했을 경우
        res = {
            "version": "2.0",
            "template": {
                "outputs": [
                    {
                        "simpleText": {
                            "text": "음식 이름이 설정되지 않았습니다. 음식 이름을 먼저 입력하세
요"
                        }
                    }
                ]
            }
        }
        return jsonify(res)

    elif food:
        youtube_api_key = [redacted]
        try:
            response = requests.get('https://www.googleapis.com/youtube/v3/search',
params={
                'part': 'snippet',
                'q': f"{food} 레시피",
                'key': youtube_api_key,
                'type': 'video',
                'maxResults': 10  # 최대 결과 수, 필요에 따라 조정
            })
            search_results = response.json()['items']

            if search_results:
                video_ids = [item['id']['videoId'] for item in search_results]

                response = requests.get('https://www.googleapis.com/youtube/v3/videos',
params={
                    'part': 'contentDetails,snippet',
                    'id': ','.join(video_ids),
                    'key': youtube_api_key
                })
                video_details = response.json()['items']
```

```python
                if video_details:
                    cards = []
                    for item in video_details:
                        video_title = item['snippet']['title']
                        video_thumbnail = item['snippet']['thumbnails']['default']
['url']
                        video_id = item['id']
                        duration = item['contentDetails']['duration']

                        # 영상 길이 측정
                        match = re.match(r'PT(\d+H)?(\d+M)?(\d+S)?', duration)
                        hours = int(match.group(1)[:-1]) if match.group(1) else 0
                        minutes = int(match.group(2)[:-1]) if match.group(2) else 0
                        seconds = int(match.group(3)[:-1]) if match.group(3) else 0
                        total_seconds = hours * 3600 + minutes * 60 + seconds

                        # 영상 길이가 60초를 넘을 때
                        if total_seconds > 60:
                            card = {
                                'title': video_title,
                                'description': 'YouTube Video',
                                'imageUrl': video_thumbnail,
                                'link': {
                                    'web': f'https://www.youtube.com/watch?v=
{video_id}'
                                }
                            }
                            cards.append(card)

                    if cards:
                        res = {
                            "version": "2.0",
                            "template": {
                                "outputs": [
                                    {
                                        "listCard": {
                                            "header": {
                                                "title": f"\"{food} 레시피\" 검색 결과"
                                            },
                                            "items": cards
                                        }
                                    }
                                ]
                            }
                        }
```

유튜브로 검색 버튼 누를 시 유튜브 API를 이용하여 검색 결과를 카드형식으로 출력(영상 길이를 60초 이상으로 지정하여 쇼츠 제외)

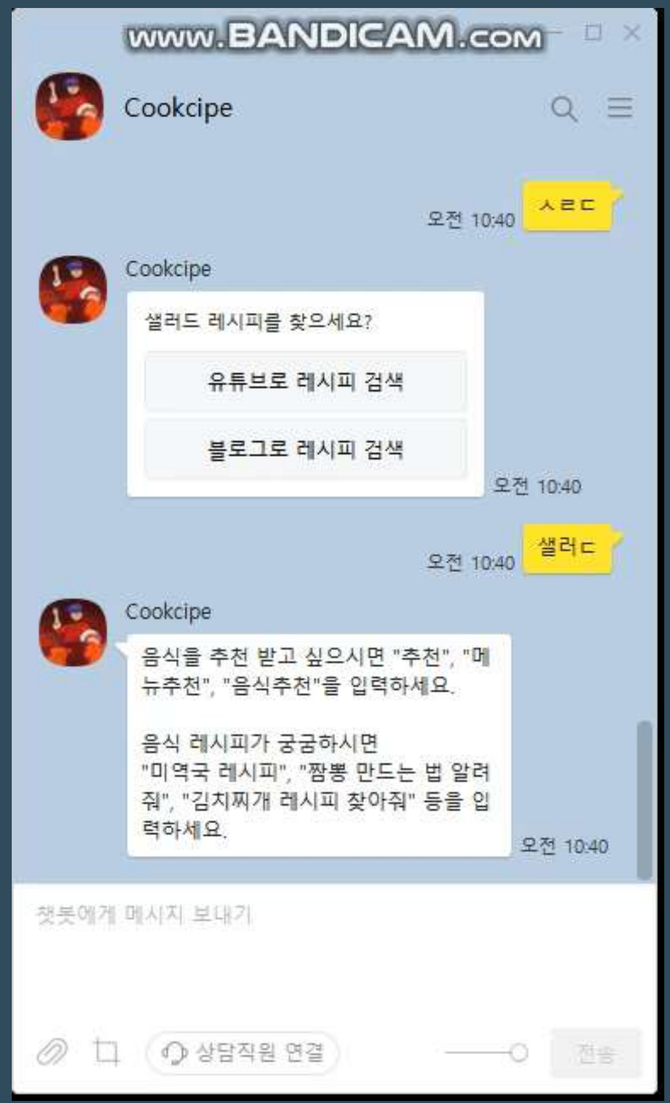
코드

```python
@application.route("/search_blog", methods=["POST"])
def searchBlog():
    global food

    if food is None:  # 채팅에 바로 '블로그로 검색'을 입력했을 경우
        res = {
            "version": "2.0",
            "template": {
                "outputs": [
                    {
                        "simpleText": {
                            "text": "음식 이름이 설정되지 않았습니다. 음식 이름을 먼저 입력하세요"
                        }
                    }
                ]
            }
        }
        return jsonify(res)

    elif food:
        naver_client_id = '[redacted]'
        naver_client_secret
        try:
            headers = {
                'X-Naver-Client-Id': naver_client_id,
                'X-Naver-Client-Secret': naver_client_secret
            }
            response = requests.get('https://openapi.naver.com/v1/search/blog.json', headers=headers,
params={
                'query': f"{food} 레시피"
            })
            search_results = response.json()['items']
```

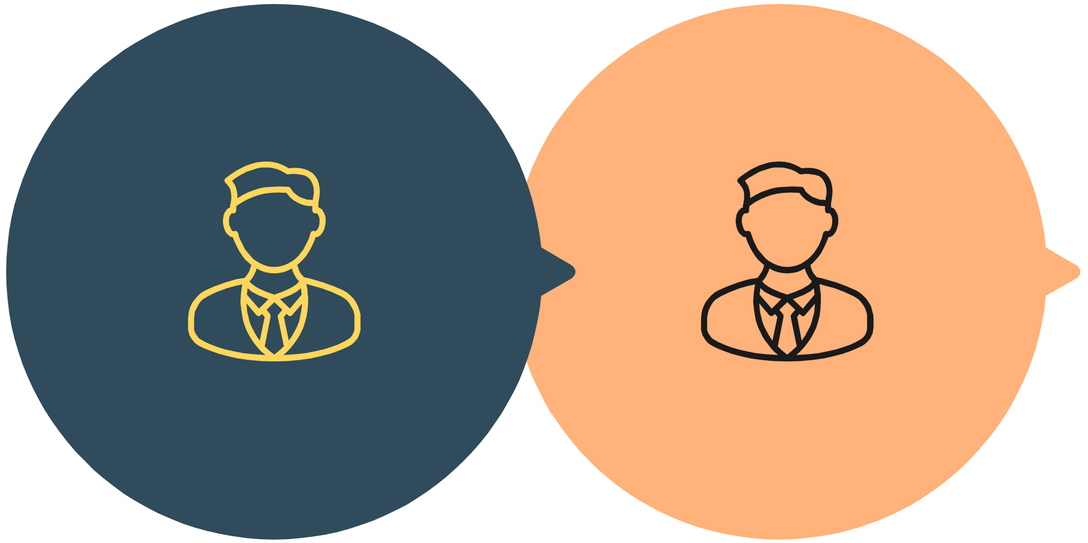
```python
if search_results:
    cards = []
    for item in search_results:
        blog_title = removeTags(item['title'])
        blog_link = item['link']

        card = {
            'title': blog_title,
            'description': 'Naver Blog',
            'link': {
                'web': blog_link
            }
        }
        cards.append(card)

    res = {
        "version": "2.0",
        "template": {
            "outputs": [
                {
                    "listCard": {
                        "header": {
                            "title": f"\"{food} 레시피\" 네이버 블로그 검색 결과"
                        },
                        "items": cards
                    }
                }
            ]
        }
    }
    food = None
    return jsonify(res)
```

블로그로 검색 버튼을 누를 시 블로그 API를 이용하여 검색 결과를 카드 형식으로 출력

시연 영상


www.BANDICAM.com
Cookcipe
오전 10:40
ㅅㄹㄷ
Cookcipe
샐러드 레시피를 찾으세요?
유튜브로 레시피 검색
블로그로 레시피 검색
오전 10:40
오전 10:40
샐러ㄷ
Cookcipe
음식을 추천 받고 싶으시면 "추천", "메뉴추천", "음식추천"을 입력하세요.
음식 레시피가 궁금하시면 "미역국 레시피", "짬뽕 만드는 법 알려줘", "김치찌개 레시피 찾아줘" 등을 입력하세요.
오전 10:40
챗봇에게 메시지 보내기
상담직원 연결
전송

자체 평가



01
김동현

처음부터 갈피를 못 잡아
어떻게 해야할 지 감이 오지 않았지만,
하나하나 천천히 하다 보니 부족하긴 하지만
결과물을 만들 수 있었고
공부가 더 필요함을 느꼈다

02
조민석

챗 봇 스킬 연동부터 어려움을 느껴 막막 했었지만
팀원의 도움으로 해결할 수 있었다.
생소하던 영역에 처음 도전해봐서 재밌었다.