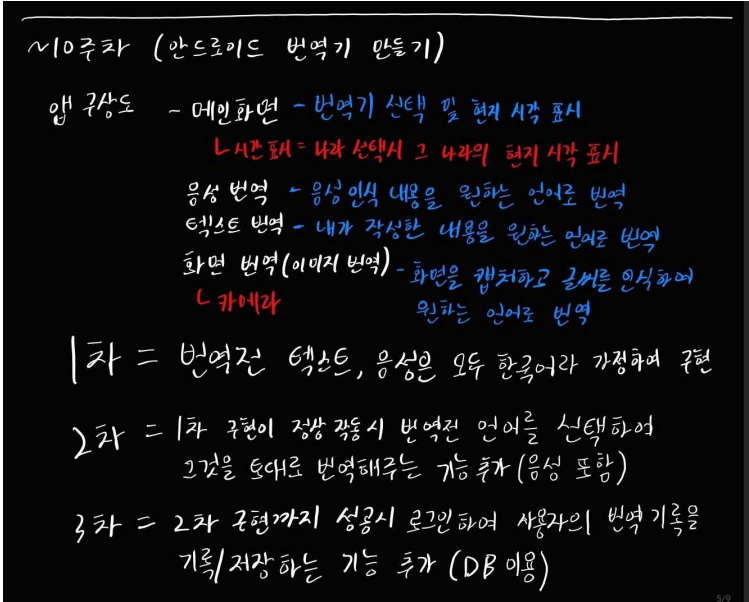
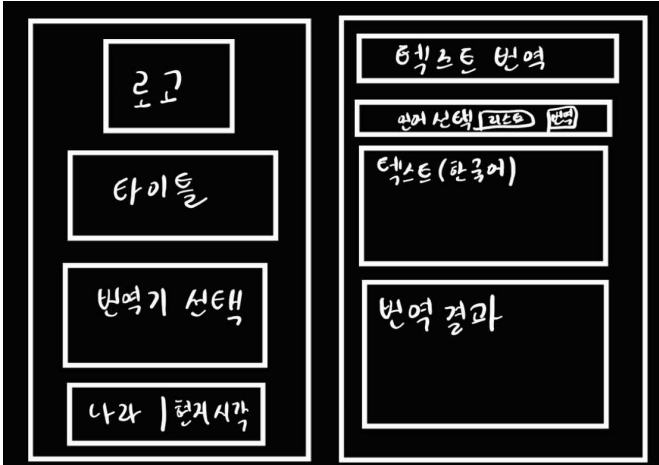
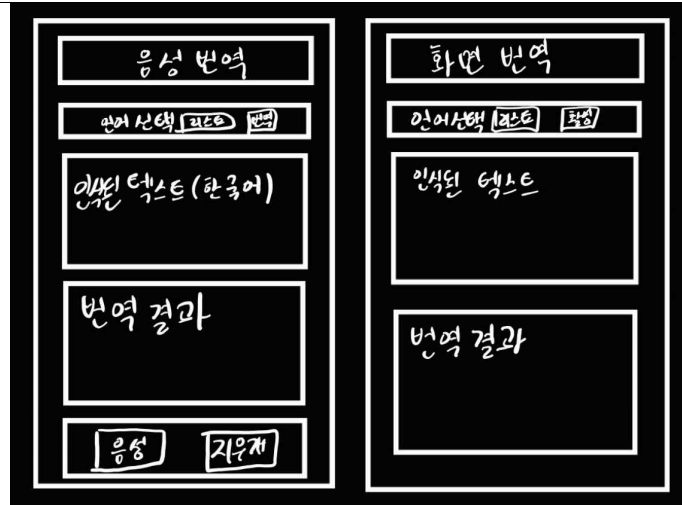


■ 서식 2 : 결과보고서 요약본

Project 결과보고서 요약본

교과목명	인공지능과 영상		
학년	3학년	이름/학번	이은우 / 202025016
프로젝트 주제	MIT App Inventor을 이용해 번역기 앱 만들기		
작품 개요	MIT App Inventor의 버튼, 레이블, 레이아웃등의 다양한 컴포넌트들을 배치하고 해당 컴포넌트들의 동작을 위해 블록코딩을 활용하여 번역기 앱을 구현한다.		
작품 구조도	<p>1. 앱 구상도 디자인</p> <ul style="list-style-type: none"> 먼저 번역기 종류와 구현 단계를 선정함.  <p>✓ 앱 안에 구현할 번역기 기능들을 선정하고, 1차 > 2차 > 3차 순서대로 구현할 것을 선정 한다.</p> <ul style="list-style-type: none"> 앱의 버튼 및 레이블 등의 배치를 구상하는 단계 		

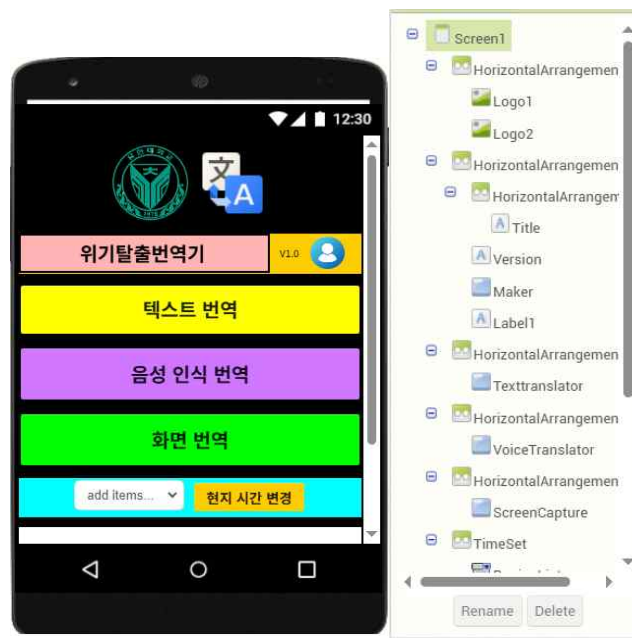


- ✓ 실제 앱 화면에서 표시될 버튼, 리스트, 텍스트 박스 등의 컴포넌트를 생각하고, 실제 동작을 위해 어디에 배치시킬지 그림으로 구상도를 그려봄.

2. 앱 디자인 & 블록 코딩

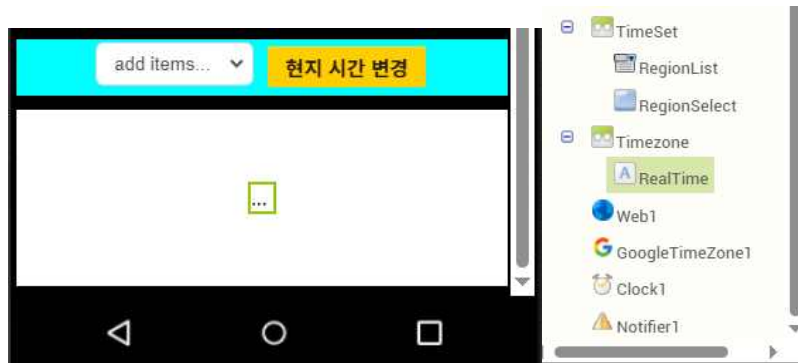
- 위에서 구성한 그림대로 앱을 디자인 함. [즉, 컴포넌트들을 배치함]

1) 메인화면[번역기 선택 & 현지 시간 표시]



- ✓ 번역기 앱의 이름은 “위기탈출번역기”로 위기상황에서 바로바로 사용하기 위한 번역기 앱이라는 컨셉으로 상황에 따라 덜 급하면 직접 텍스트를 작성하는 ‘**텍스트 번역기**’, 상대방의 말을 못 알아먹을 때 급하게 사용해야할 때 음성을 인식하고 그것을 한국어로 번역하는 ‘**음성 인식 번역기**’, 그리고 음식점에서 메뉴판이나 호텔에서 안내 문구를 해석하기 위해 카메라를 사용한 ‘**화면[이미지]번역기**’ 등으로 이동할 수 있는 버튼을 구현함.
- ✓ 그리고 추가적인 기능으로 최하단에 여행하는 그 나라의 실제 현지 시각을 알기 위한 시계를 표현하는 시계를 구현한다.

1.1. 문제점



✓ 현지 시간을 표시하기 위한 시계의 기능적인 문제인데, 여행한 나라의 현지 시간을 알기 위해 앱에서 시간을 표시하는 방법으론 2가지가 존재했다.

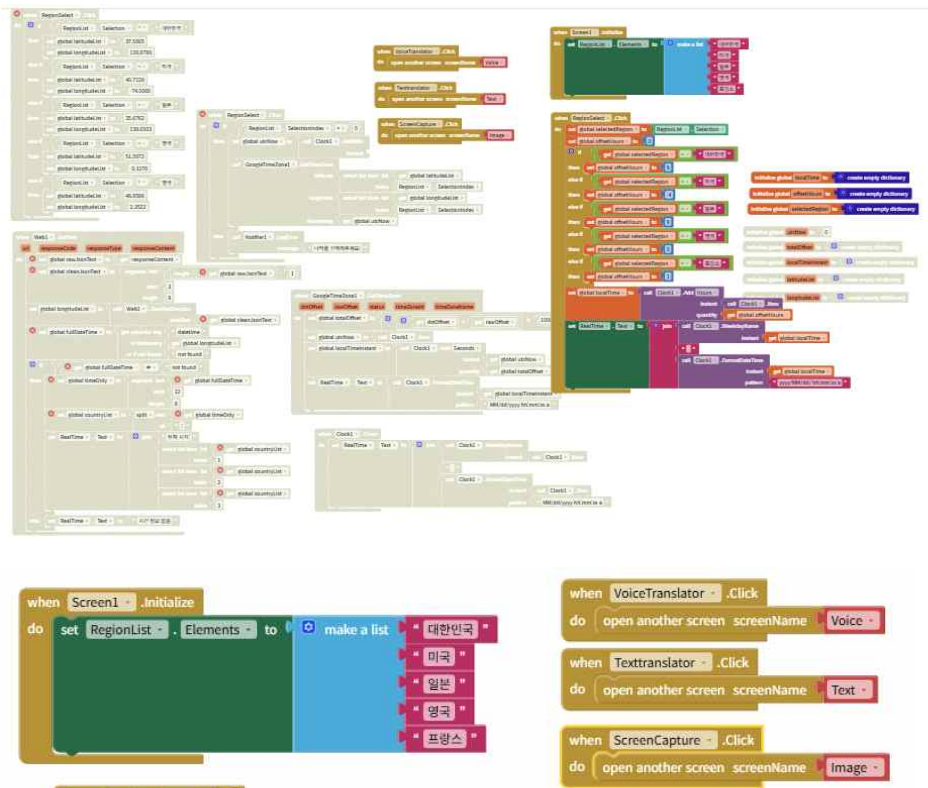
① 선택한 나라에 따라 시차를 적용한 시간을 계산하는 [수동 방식]

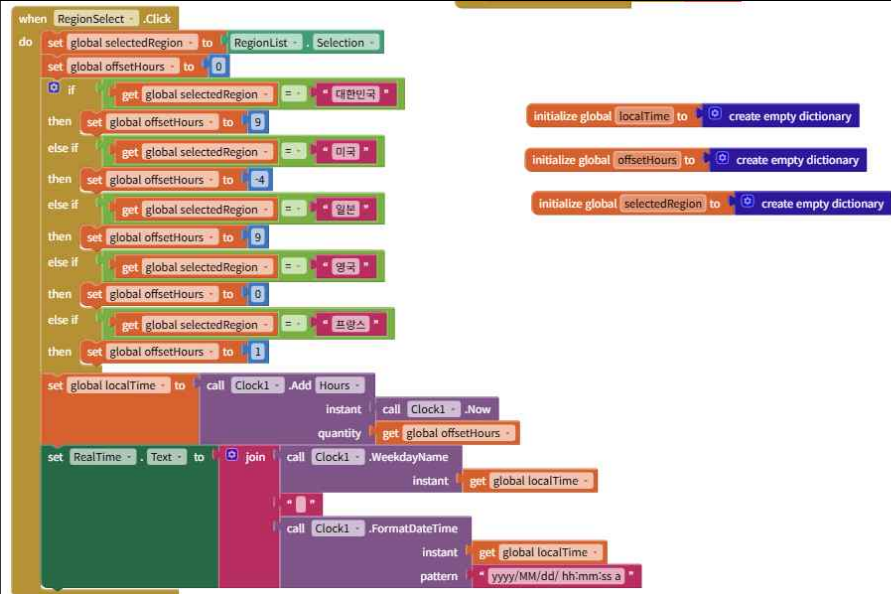
② 나라를 선택하면 그 나라 지역의 위도/경도를 기반으로 [Google Time Zone API]를 요청하는 방식

✓ 2번의 경우엔 API 사용 후 나온 결과가 JSONText 형식인데 이를 Decode[복호화]까지 해야 하는 복잡한 방식으로 설계되고 JSONText의 형태가 API별로 다르게 구성되어있어 정확하게 복호화하지 못하는 오류가 자주 발생하여 가장 쉬운, 하지만 각 나라의 시차를 긴 시간의 작업 방식을 택하였다.

▶ 고려사항

☞ 컴포넌트 중에 [Maker]라는 버튼이 존재한다. 본래는 만든 제작자 소개를 위한 화면을 구현하려 했으나, 이를 클릭하면 유저가 로그인하는 화면을 구상하고, 로그인해서 사용했던 번역, 번역 기록을 저장하는 기능을 구현하는 것을 목표로 한다.





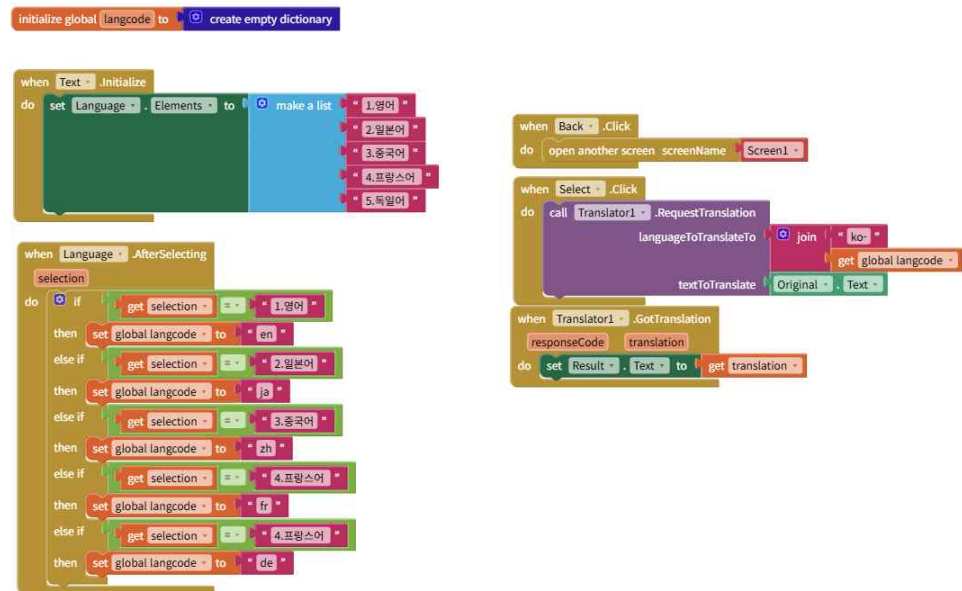
2) 텍스트 번역 화면[텍스트를 입력하여 원하는 나라 언어로 번역]



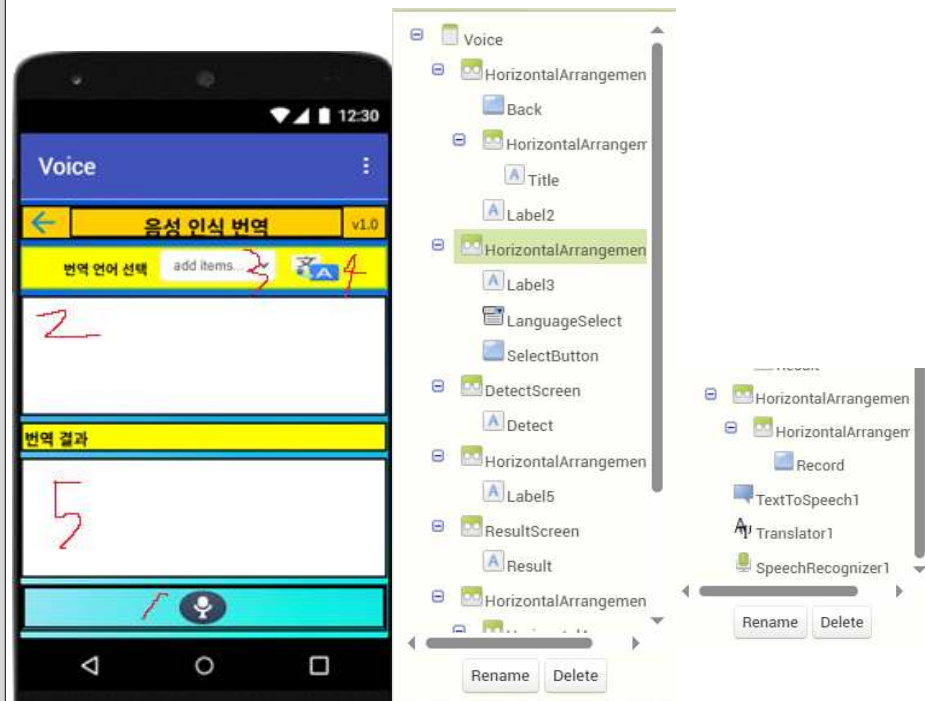
- ✓ 메인화면에서 텍스트번역 버튼을 누르면 나오는 화면이다.
해당 화면에선 왼쪽 상단에 파란 화살표 버튼[Back]을 누르면 다시 메인화면으로 돌아가며, 1번의 텍스트 박스[original]에 한국어를 입력후, 2번의 번역될 언어를 선택하는 리스트[Language]가 있다. 번역할 언어를 선택하고 3번인 구글번역 아이콘 이미지 버튼[Select]을 누르면 4번의 레이아웃의 레이블[Result]에 번역된 결과가 나오게 된다.

▶ 고려사항

- 해당 번역 기능은 GoogleTranslator API를 사용하는데, API마다 제한된 Key값을 제공하므로 번역하는데에 한계가 있다. 그러므로 이 API Key를 새로 발급하는 자동화 기능을 구현함
- 1차 구현단계인 번역할 원본 언어가 한국어로 강제되어있는데 정상적으로 테스트가 되었고, 번역에 문제가 생기지 않으면 꼭 한국어만이 아닌 다른 나라 언어로 선택 후 번역하고 싶은 나라의 언어로 번역하는 기능도 구현함



3) 음성 인식 번역[음성을 인식하여 원하는 나라 언어로 번역]



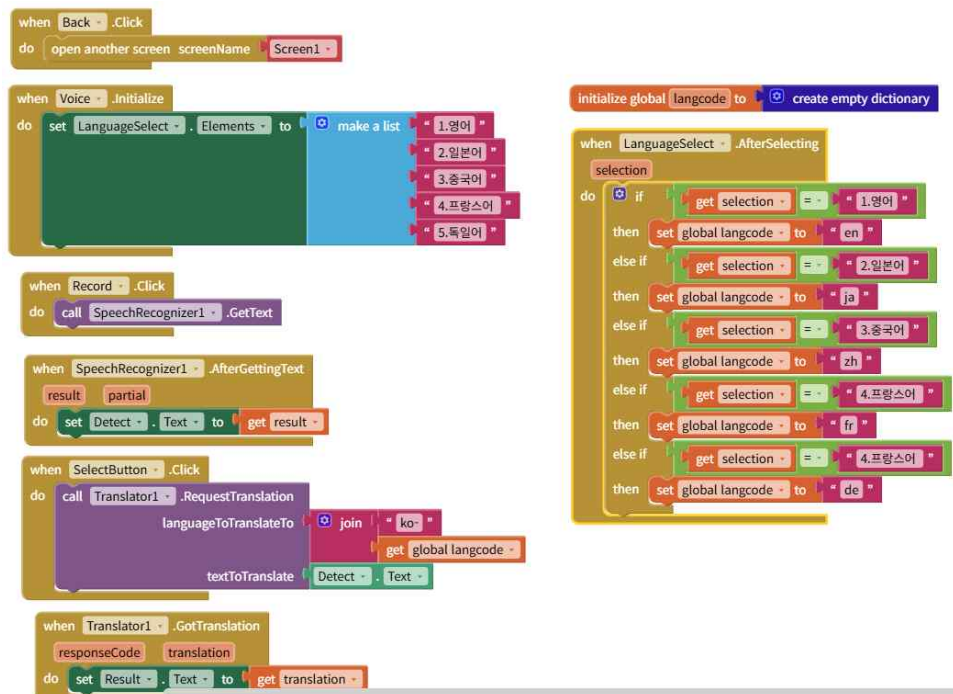
✓ 메인화면에서 음성 인식 번역 버튼을 누르면 나오는 화면이다.

왼쪽 상단에 파란 화살표 버튼[Back]기능은 동일하고, 1번의 마이크 아이콘 버튼 [Record]을 누르면, 앱에 내장된 컴포넌트인 SpeechRecognizer를 통해 녹음을 하고, 2번 박스안에 존재하는 레이블[Detect]에 인식된 음성을 앱에 내장된 컴포넌트인 TextToSpeech를 통해 텍스트로 표현한다. 이후 3번에 번역하고 싶은 나라[LanguageSelect]를 고르고 4번의 구글번역기 아이콘 버튼[SelectButton]을 클릭 시 텍스트 번역과 마찬가지로 Google Translate API를 이용하여 번역하고 5번 박스 안에 존재하는 레이블[Result]에 번역된 결과를 출력한다.

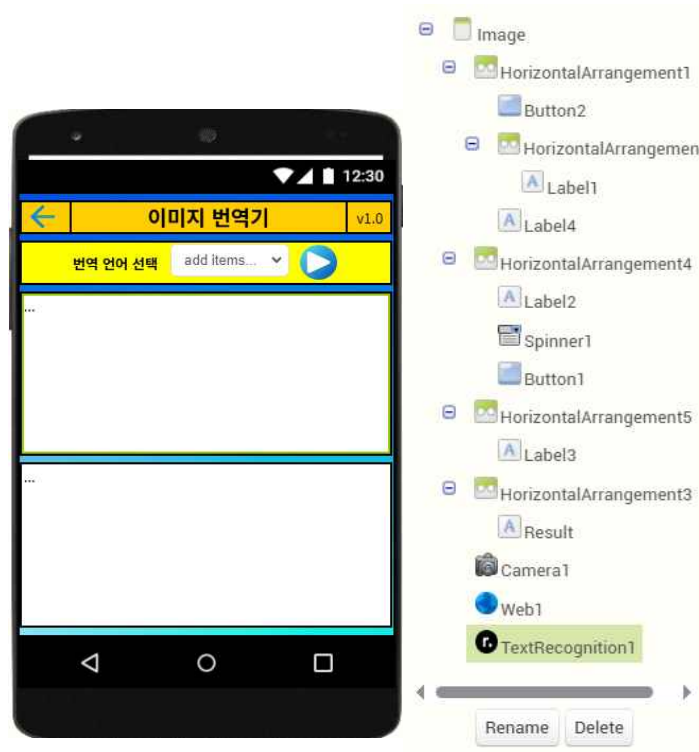
▶ ※고려사항※

☞ 여기도 텍스트 번역기와 마찬가지로 GoogleTranslateAPI를 사용하기에 API마다 제한된 Key값으로 번역이 진행된다. 이 API Key를 새로 발급하는 자동화 기능을 구현함.

☞ 두 번째도 비슷하다. 현재는 인식되는 음성이 한국어로 고정되어있으나, 이번에는 음성 인식인 만큼 인식된 음성의 언어를 자동으로 분석하여 언어를 파악하는 기능을 구현함.



4) 화면 인식 번역[카메라 기능으로 사진의 텍스트를 분석후 번역]



- ✓ 원하는 번역 언어를 선택하고, 재생 버튼으로 사진을 찍고 Web 컴포넌트를 통해 이미지에 있는 텍스트를 기계가 읽을 수 있는 디지털 텍스트로 변환하는 기술 OCR API를 호출하여 TextRecognition을 통해 번역을 진행한다.

▶ ※문제점※

- ✓ 사용할 API의 서버 문제, API Key의 한도/제약 등으로 적합한 API를 찾는 것이 어렵기도 하고, 새로 찾더라도 이 API들의 출력문이 JSON text 형태인데 포맷이 모두 달라 근본적으로 이 부분을 해결 해야함.
API를 사용하지 않고 하기엔 어려운 부분이 이미지를 분석하는 OCR API와 번역을 하는 번역기 API 총 2개가 이용되어야 하므로 아마도 적합한 API를 찾는 것이 문제해결의 우선순위가 될 것 같다.

결과물 제작

Translator.apk 안드로이드 앱으로 Build 함

프로젝트를
통해 느낀점

비록 완벽한 애플리케이션 제작은 아니었지만, 코딩 능력이 뛰어나지 않아도, '노코드' 즉 여기서 블록 프로그래밍을 통해 한눈으로 쉽게 동작 과정을 짤 수 있고, 그 외 다양한 확장프로그램을 이용하여 쉽게 앱을 만들 수 있다는 점에서 신기한 경험이었다. 다양한 문제점들이 프로젝트 진행에 방해가 되지 만 이 수업이 끝나고도 계속해서 문제를 분석하고 해결함으로써 기술 활용에 좋은 발전 경험을 쌓을 예정이다. 또 다른 프로젝트에서 간단하게 앱을 만들 수 있는 프론트 엔드 개발 경험으로 활용할 수 있는 기술이 늘어난 좋은 기회였다.