

# PC 간 자동 폴더 동기화 어플리케이션

Progress report – 1

## 목차

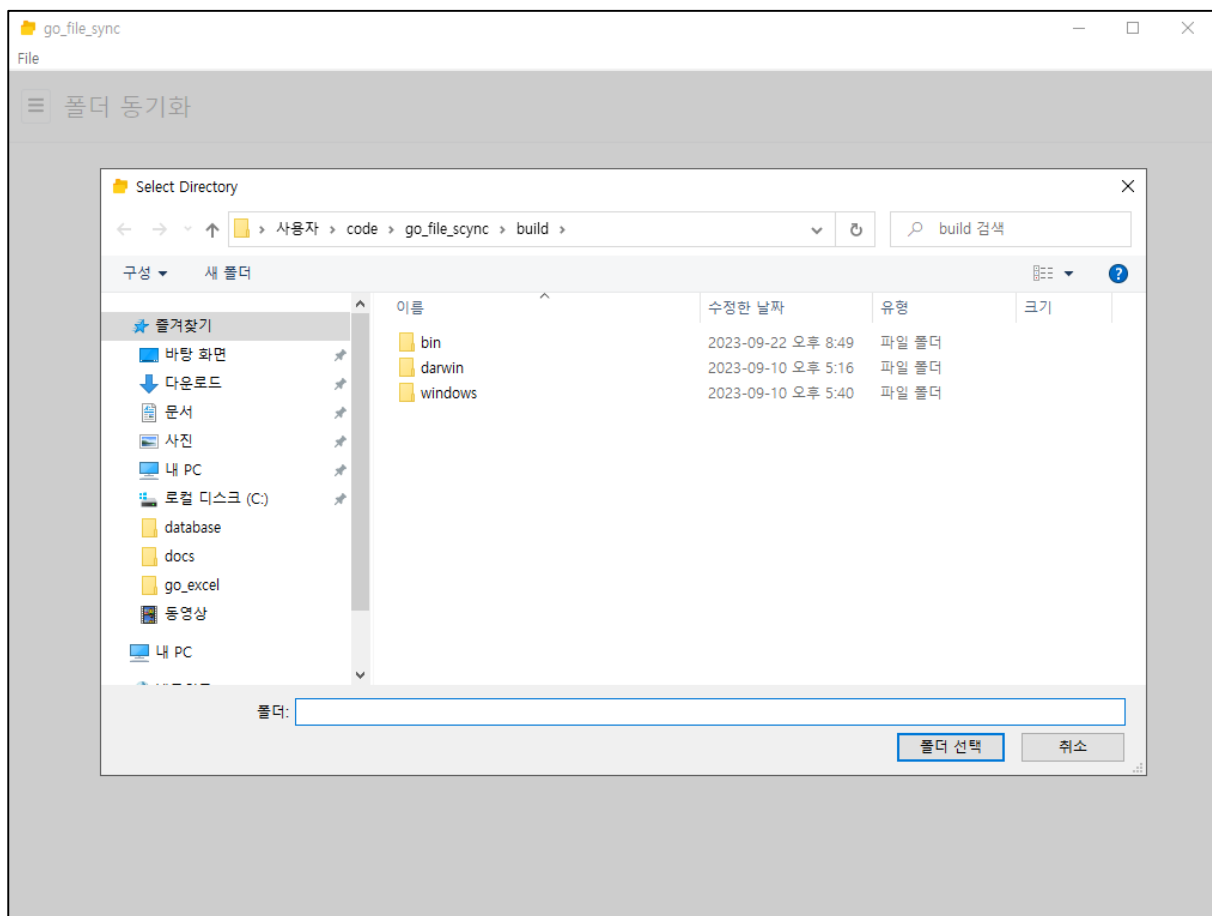
PC 간 자동 폴더 동기화 어플리케이션 .....	1
Progress report – 1 .....	1
개발 내용 .....	3
메인 페이지의 UI 개발 .....	3
프로그램 실행 시 최초의 화면 .....	3
로딩 화면 .....	4
메인 화면 .....	5
동기화 폴더 선택 시에 폴더 안에 있는 모든 정보 가져오기 .....	7
폴더 탐색 .....	7
파일 및 폴더 정보 수집 .....	7
공부한 사항 .....	8
파일 및 폴더 구조 이해 .....	8
이벤트 처리와 비동기 프로그래밍 .....	8
당면한 어려움 .....	9
폴더 구조 시각화 .....	9
재귀적인 구현 .....	9
사용자 경험 고려 .....	9
느낀 점 .....	10
사용자 중심의 생각 .....	10
효율적인 파일 시스템 탐색 .....	10

## 개발 내용

### 메인 페이지의 UI 개발

메인 페이지의 UI를 개발하여 프로젝트의 기본 레이아웃과 사용자 인터페이스를 구현했습니다. 이 부분에서는 디자인과 사용자 경험에 대한 고민을 많이 했는데, 사용자 편의성을 고려하면서도 동시에 프로젝트의 주요 목적인 폴더 동기화 기능을 잘 부각시키는 것이 중요했습니다.

### 프로그램 실행 시 최초의 화면

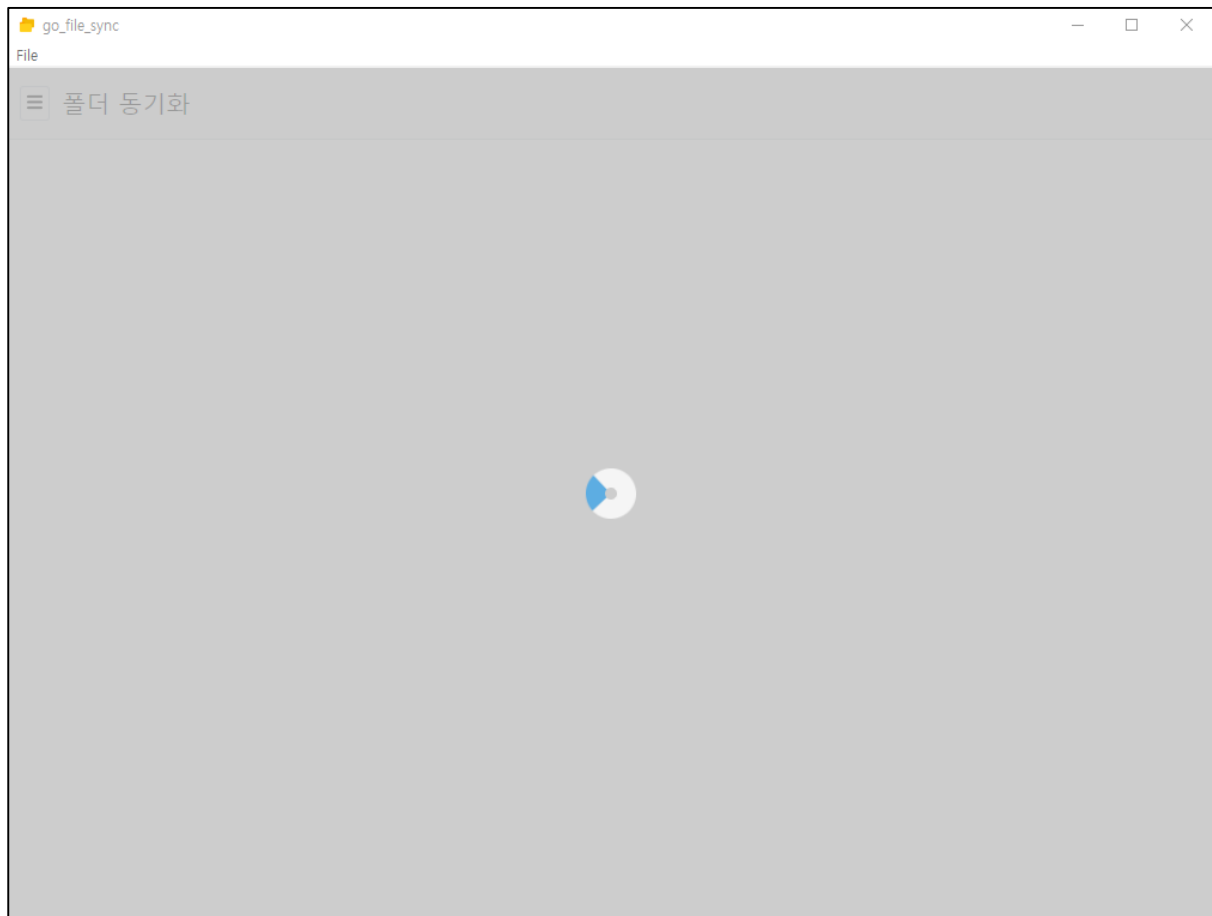


사용자가 어플리케이션을 실행하면 다이얼로그 창이 열립니다.

다이얼로그는 사용자에게 “폴더 선택”을 요청하는 메시지와 함께 “폴더 선택” 버튼을 클릭하여 자신이 동기화하고자 하는 폴더를 선택할 수 있습니다.

폴더 선택 다이얼로그 이후에는 폴더를 재귀적으로 돌며 파일들의 정보를 가져오는 작업을 수행하기 전에 로딩 화면이 나타나는데, 이 로딩 화면은 사용자에게 작업이 진행 중임을 알려줍니다.

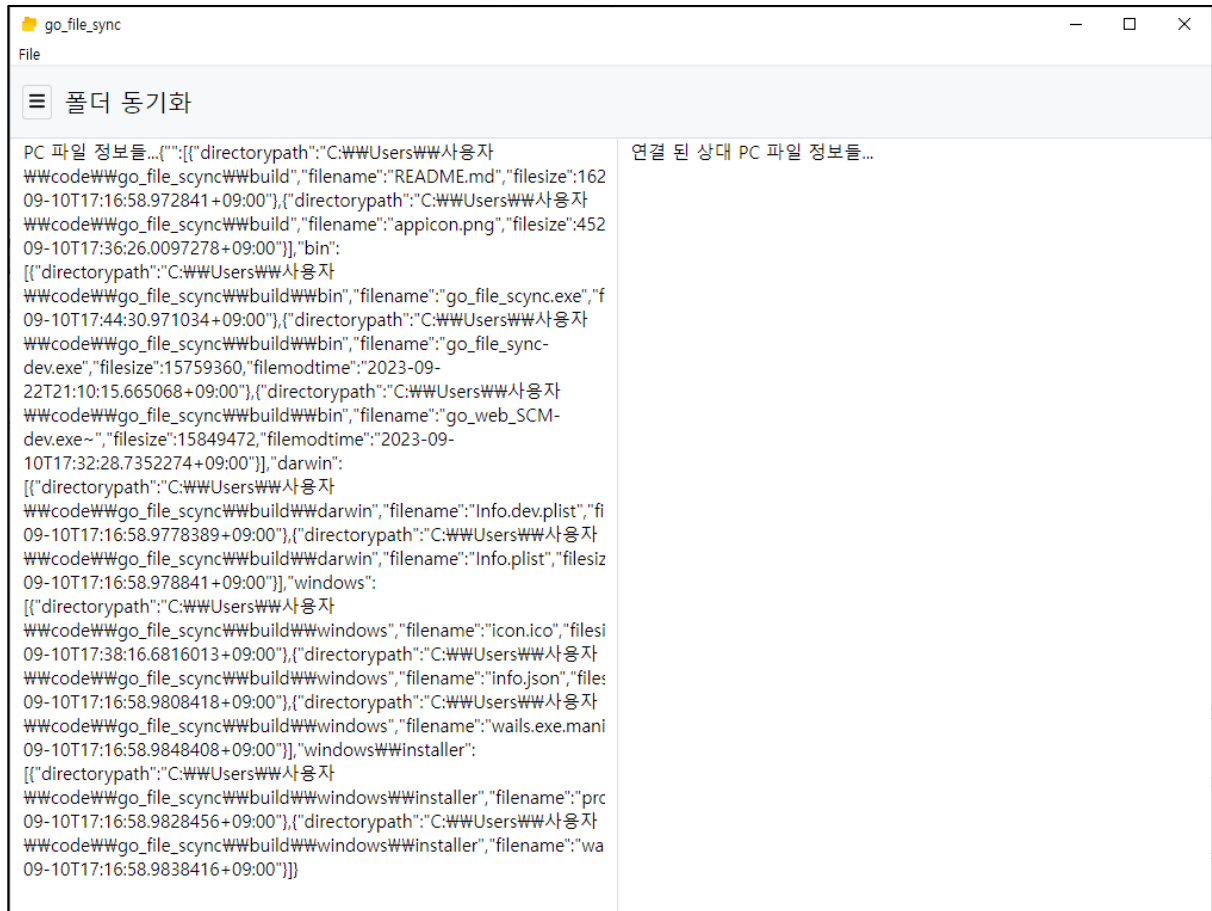
## 로딩 화면

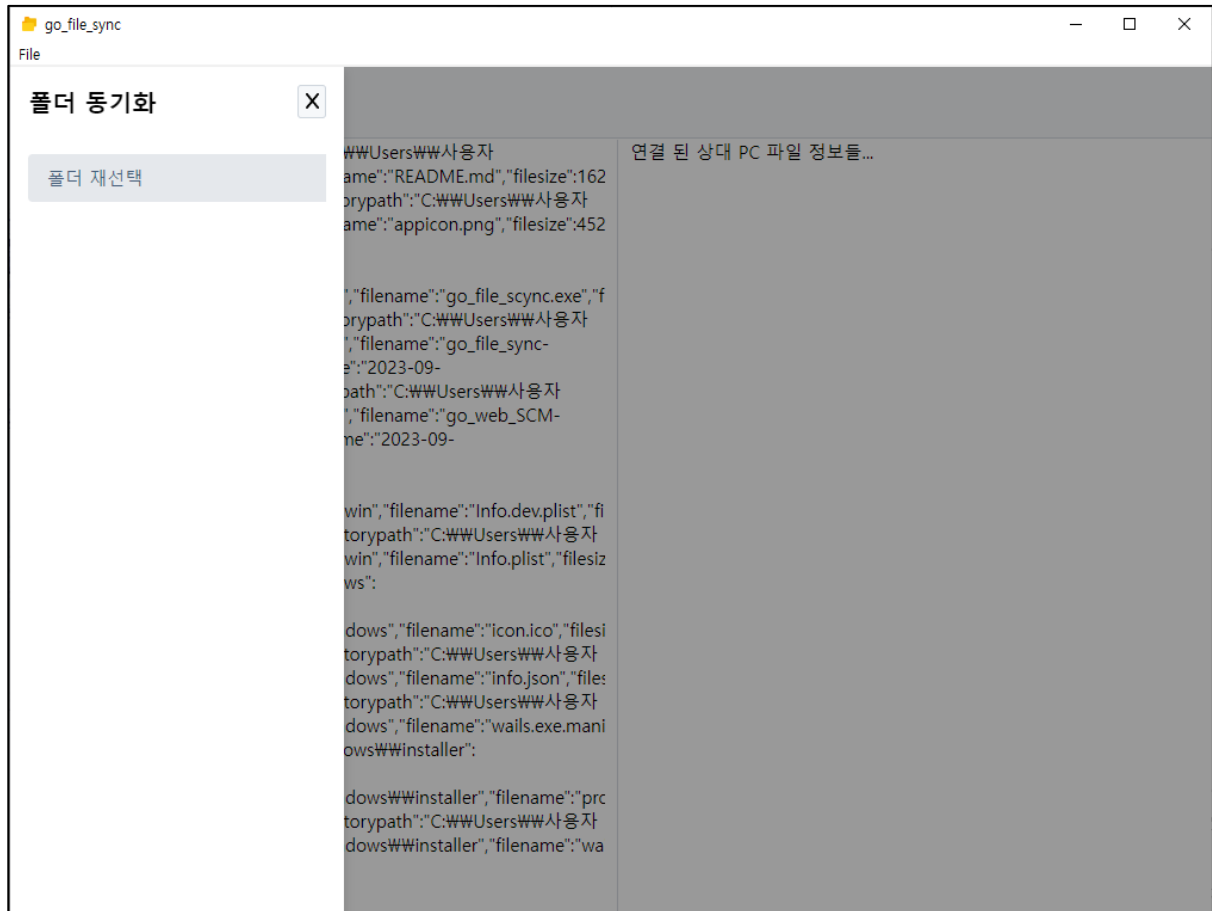


로딩 화면에는 로딩 스피너(로딩 동그라미)가 표시됩니다. 이 스피너는 작업이 진행 중임을 시각적으로 나타내며, 사용자에게 기다림을 전달합니다.

로딩 화면을 통해 사용자는 폴더 동기화 작업의 진행 상황을 파악하고, 작업이 완료될 때까지 기다릴 수 있도록 도움을 받게 됩니다.

## 메인 화면





메인 화면으로 넘어왔을 때 Navbar와 선택된 폴더 안의 파일 정보를 텍스트로 확인할 수 있는 기능이 구현되어 있습니다.

현재 Navbar는 어플리케이션의 동기화 폴더 재선택만이 존재하며 나중에 사용자의 편의성을 더 하는 기능을 추가할 예정입니다.

현재 선택된 폴더 안에 있는 파일 및 폴더에 대한 정보가 텍스트 형식으로 제공됩니다. 사용자는 이 정보를 확인하여 동기화할 항목을 살펴볼 수 있습니다.

파일 및 폴더에 대한 정보는 나중에 폴더 구조 형식으로 변경할 예정입니다.

사용자가 파일과 폴더 간의 관계를 더 쉽게 이해하고 탐색할 수 있을 것입니다.

동기화 폴더 선택 시에 폴더 안에 있는 모든 정보 가져오기

사용자가 동기화할 폴더를 선택할 때, 폴더 안에 있는 모든 파일과 하위 폴더의 정보를 가져오는 기능을 구현했습니다. 이 과정에서 파일 시스템을 탐색하고, 정보를 수집하는 부분에서 파일 구조 알고리즘에 대한 이해가 필요했습니다.

#### 폴더 탐색

선택한 폴더 내에는 다른 하위 폴더와 파일이 존재할 수 있으므로, 재귀적으로 모든 폴더를 탐색합니다. 이때, 각 폴더와 파일에 대한 정보를 수집합니다.

#### 파일 및 폴더 정보 수집

각 폴더와 파일에 대한 정보를 수집합니다. 이 정보에는 파일 이름, 크기, 수정일, 경로 등이 포함됩니다.

## 공부한 사항

### 파일 및 폴더 구조 이해

프로젝트를 개발하면서 파일 및 폴더의 구조와 관련된 핵심 개념을 숙지하였습니다. 이를 통해 파일 및 폴더를 효과적으로 탐색하고 관리하는 방법을 배웠습니다.

### 이벤트 처리와 비동기 프로그래밍

사용자의 파일 및 폴더 조작에 대한 이벤트 처리와 비동기 프로그래밍을 공부하였습니다. 이를 통해 프로그램이 여러 작업을 동시에 처리하고 사용자 경험을 개선하는 방법을 습득하였습니다.



## 당면한 어려움

이번 주에 느낀 가장 큰 어려움은 선택된 파일 및 폴더 구조를 화면에 효과적으로 표시하는 것이었습니다. 파일과 폴더를 가져오는 로직은 성공적으로 구현했지만, 이 정보를 사용자 친화적인 폴더 구조로 시각화하여 표현하는 데 어려움을 겪었습니다.

### 폴더 구조 시각화

선택한 파일 및 폴더의 구조를 사용자에게 직관적으로 시각화하여 보여주는 것이 중요했습니다. 그러나 이를 구현하는 데 필요한 트리뷰(TreeView)나 폴더 아이콘, 드래그 앤 드롭 등의 기능을 어떻게 구현해야 할지에 대한 고민이 있었습니다.

### 재귀적인 구현

폴더 내에 폴더가 중첩되어 있을 경우, 재귀적으로 모든 하위 폴더를 탐색하고 표시해야 했습니다. 이 과정에서 중복되는 목록을 효과적으로 관리하는 것이 어려웠습니다.

### 사용자 경험 고려

사용자가 폴더 구조를 쉽게 이해하고 조작할 수 있도록 UI/UX 디자인에도 고민을 많이 했습니다. 각 폴더와 파일을 어떻게 시각화 하고, 사용자에게 어떤 상호작용을 제공할지 고려해야 했습니다.

## 느낀 점

### 사용자 중심의 생각

사용자가 프로그램을 어떻게 사용하며 어떤 어려움을 겪는지에 대한 이해가 중요하다는 것도 깨달았습니다. 사용자의 요구를 충족시키고 사용자 경험을 개선하는 것이 프로젝트의 핵심이라는 것을 깨달았습니다.

### 효율적인 파일 시스템 탐색

Go 언어를 사용하여 파일 시스템을 재귀적으로 탐색하고, 복잡한 폴더 구조를 비교적 간단하게 탐색하고 정보를 수집할 수 있습니다.