2025-2 기초웹개발론

FE 리스크 관리 문서

컴퓨터교육과 2021104133 이우령

초기 UI 구성 및 환경 설정	
리스크 내용	크롬 브라우저 및 크롬 확장 프로그램 환경에서는 TypeScript 파일을 직접 실행하거나 로드할 수 없다는 제한이 있다. 개발 초기 단계에서 .ts 파일을 그대로 사용하려고 시도하면서 스크립트가 실행되지 않는 문제가 발생했고, 이로 인해 개발 진행이 일시적으로 중단되었다.
해결 방법	tsc를 사용해 js로 컴파일하여 크롬에서 로드하도록 개발 진행하려고 하였으나, ts 경험자가 적어 js로 개발을 진행하는 것이 진행에 용이할 것으로 판단하여 js로 개발을 진행하였다.
느낀 점	이번 문제를 통해 브라우저 실행 환경과 개발 언어 간의 차이를 명확히 이해하게 되었다. TypeScript는 개발 생산성과 안정성을 높여주지만, 실제 실행 단계에서는 반드시 JavaScript로 변환되어야 한다는 점을 간과하면 개발 초기에 불필요한 시행착오가 발생할 수 있음을 깨달았다. 앞으로는 개발 환경과 실행 환경을 구분하여 기술 스택을 설계하는 것이 중요하다는 교훈을 얻었다.

주식 위젯	
리스크 내용	네이버 증권 페이지에서 종목 정보를 수집하는 과정에서 Stock ID 값을 추출할 때, 알파벳과 숫자가 함께 포함된 ID임에도 불구하고 숫자 부분만 인식되어 가져오는 문제가 발생하여 알파벳이 포함된 ID의 주식 차트를 가져오지 못하는 문제가 발생하였다. 이로 인해 일부 해외 주식이나 특정 종목의 식별자가 불완전하게 저장되어 데이터 매칭 오류가 발생할 가능성이 있었다.
해결 방법	기존 로직이 정규식 또는 파싱 과정에서 숫자만을 대상으로 처리하고 있음을 확인하였다. 이를 해결하기 위해 Stock ID 추출 시 알파벳과 숫자를 모두 포함할 수 있도록 정규식을 수정하고, 문자열 전체를 대상으로 ID를 파싱하는 로직으로 개선하였다. 이를 통해 영문 코드가 포함된 Stock ID 또한 정상적으로 인식하고 저장할 수 있도록 구현하였다.

느낀 점	이번 문제를 통해 외부 서비스의 데이터 구조를 단순화하여 가정하는 것은 위험할 수 있다는 점을 깨달았다. 특히 금융 데이터처럼 다양한 형식의 식별자가 존재하는 경우, 초기 설계 단계에서 데이터 형식의 확장 가능성을 고려하는 것이 중요하다는 교훈을 얻었다. 앞으로는 파싱 로직을 구현할 때 입력 데이터의 다양한 패턴을 사전에 검토하고 대응하도록 할 계획이다.

비동기 처리로 인한 초기 렌더링 오류	
리스크 내용	이드바 초기화 과정에서 chrome.storage.sync.get()을 통해 설정값과 위젯 상태를 불러오도록 구현하였다. 그러나 Chrome Storage API는 비동기 방식으로 동작하기 때문에, 저장된 설정값이 모두 로드되기 전에 UI 렌더링이 먼저 실행되는 문제가 발생할 수 있었다. 이로 인해 위젯 순서가 초기값으로 표시되거나, 비활성화된 위젯이 일시적으로 노출되는 현상이 발생하였다.
해결 방법	스토리지에서 불러온 데이터를 기본값과 병합한 뒤, 위젯 순서를 normalizeWidgetOrder()로 정규화하는 과정을 선행하도록 로직을 정리하였다. 또한 저장된 widgetOrder가 없을 경우 초기값을 저장하도록 처리하여 다음 실행 시 동일한 상태가 유지되도록 개선하였다.
느낀 점	Chrome Extension 환경에서는 비동기 API 사용이 필수적인 만큼, 데이터 로딩 순서와 UI 렌더링 타이밍을 명확히 분리하는 것이 중요하다는 점을 깨달았다. 초기화 단계에서의 작은 비동기 처리 실수도 사용자 경험에 직접적인 영향을 줄 수 있기 때문에, 앞으로는 상태 준비가 완료된 이후에만 UI를 렌더링하는 구조를 더욱 명확히 설계할 계획이다.