

[서식8] 산학연계 캡스톤디자인 결과보고서

## 2025학년도 산학연계 캡스톤디자인 과제 결과보고서

과제 현황					
과제명	버전 관리 GUI 프로젝트				
팀명	MOBABI				
과제 수행기간	2025년 3월 ~ 2025년 11월				
Github주소	<a href="https://github.com/HBNU-SWUNIV/mobile-capstone25-mobile-capstone-mobabi">https://github.com/HBNU-SWUNIV/mobile-capstone25-mobile-capstone-mobabi</a>				
지 도 교 수		참여기업(멘토)			
소 속	모바일융합공학과	기 업 명	(주) 에셈블		
성 명	박현주	성 명	서민석		
전 화	010-9953-1220	전 화	010-6424-8324		
E-mail	phj@hanbat.ac.kr	E-mail	msseo@assemble.com		
참 가 인 원(팀원)					
No.	성 명	소속학과	학 번	전화번호	E-mail
1	이은채	모바일융합공학과	20221032	010-3066-0747	2snha@naver.com
2	김나연	모바일융합공학과	20221006	010-572209568	kny7024@naver.com
3					
4					
5					
6					
7					
8					

### 과제 세부 설명

소프트웨어 개발 과정에서 버전 관리 시스템은 필수적인 도구이다. 특히 Git은 가장 널리 사용되는 버전 관리 시스템 중 하나로, 코드 변경 사항을 추적하고 협업을 원활하게 수행할 수 있도록 도와준다.

하지만 현재 Git은 CLI(Command Line Interface)를 중심으로 동작하는 경우가 많기 때문에, 개발을 시작한 지 얼마 안 된 초보 개발자나 비전문가들은 사용에 다소 어려움을 느낄 수 있다.

CLI 기반의 Git 사용이 가지는 주요 한계는 다음과 같다.

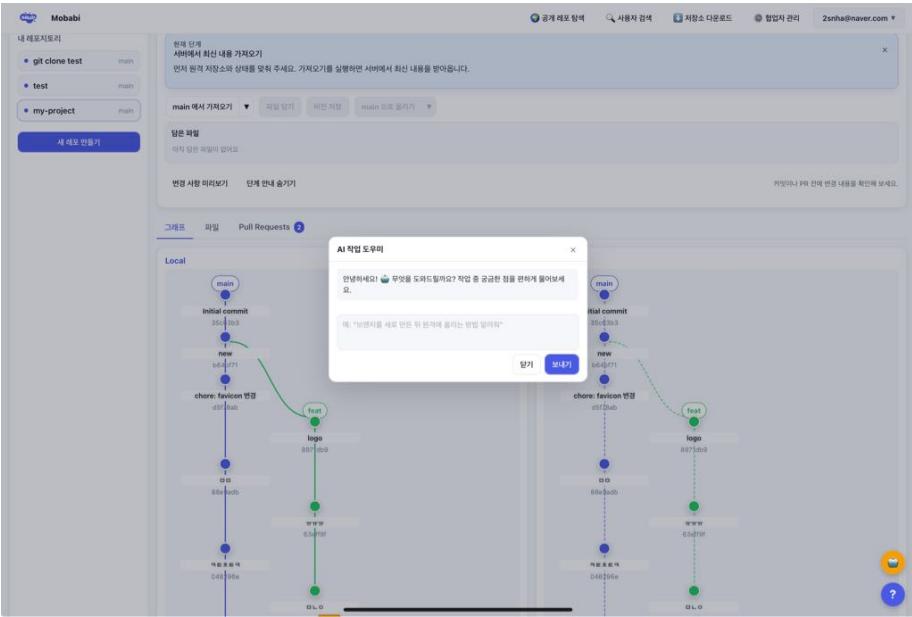
- 높은 진입 장벽: 다양한 명령어와 옵션을 암기해야 하며, 사소한 실수로 인해 프로젝트에 손상이 발생할 가능성이 있음
- 직관적인 이해가 어려움: 브랜치 구조나 커밋 히스토리를 텍스트 기반 명령어로만 확인해야 함
- 협업 과정에서 실수 가능성 증가: 명령어 입력 실수 등으로 인해 의도하지 않은 변경 사항이 발생할 수 있음

#### 1. 개발 동기 및 목적, 필요성

이러한 문제는 단순히 초보 개발자에게만 국한되지 않고, 기업 환경에서도 비효율적인 요소로 작용할 수 있다.

- 대규모 협업 프로젝트에서 코드 관리가 더욱 복잡해짐
- Git 사용법에 미숙한 팀원이 실수로 코드베이스에 문제를 일으킬 가능성 증가
- 신규 개발자 온보딩 과정에서 Git 사용법 교육에 많은 시간이 소요됨

	<p>백엔드는 가장 기본적인 add, commit, push 기능을 구현한다. 파일 시스템(I/O) 접근을 통해 변경 사항을 수집하고, 이를 특정 형태로 압축 및 저장하여 로컬 또는 서버로 전송할 수 있도록 한다. 또한 프론트엔드와의 원활한 소통을 위해 RESTful API, 명확한 엔드포인트, 스펙 문서화 등을 통해 API를 구성한다.</p> <p>프론트엔드에서는 Vite와 React, JavaScript를 사용하여 구현한다. 한 화면 내에서 처리해야 할 요소가 많으므로 커밋 내역 리스트 컴포넌트, 브랜치 현황 컴포넌트, 기타 주요 기능 수행 컴포넌트 등으로 모듈화하여 구현한다. 또한 useState, useEffect 등을 활용해 현재 선택된 브랜치, 변경된 파일 목록, 커밋 로그 상태 등을 체계적으로 관리하고, 백엔드에서 받은 API를 호출하여 실시간 상태를 반영하여 렌더링할 수 있게 한다.</p> <p>디자인은 Figma로 구성하여 프론트엔드로의 변환을 쉽게 하였으며, 다양한 사용자 환경에서 데스크탑을 주로 사용할 것으로 예상되어 웹 사이트 최적화 비율로 디자인하였다.</p> <p>먼저 기존 사용자들이 가장 많이 사용하는 라이브러리를 활용한다. 공식 문서를 참조하여 주요 기능 호출 시 반환되는 응답과 프론트엔드 처리 방법을 파악한다. 이후 프론트엔드와 백엔드 모두 간단한 테스트 프로젝트로 각 API의 완성도를 검증한 뒤 실제 연동이 가능한지 확인한다. 이 과정을 통해 전체 아키텍처의 구성 가능성을 미리 점검하고 완성도를 높인다.</p> <p>이후 GitHub API를 사용하던 부분을 자체 API로 대체하고 동일한 요청이 정상 처리되는지, UI에 문제가 없는지 등을 추가로 테스트한다. 최종적으로 사용자가 원격 저장소에서 코드를 가져오고(Pull), 로컬의 코드를 추가(Add)하고, 원격 저장소에 올리고(Commit), 최종적으로 원격 저장소에 반영하는(Push) 과정을 자체적인 API로 구현한다. 데이터를 저장소에 반영하며, 프론트엔드와 완벽히 연동되는 것을 최종 목표로 한다.</p>
3. 과제 수행 결과	<p>가장 큰 진입장벽인 언어 설명부터, 사용자와 밀접하게 연계되어 직접 소통하며 문제점을 파악해나가는 AI 챗봇 서비스까지 모두 구현에 성공하였다.</p> <p>실제로 저장소에 파일을 올리거나 내려받는 기능, 내 저장소에 팀원을 초대하여 서로의 코드를 확인하고 리뷰를 달며 의견을 공유하는 협업 시스템도 완벽하게 구축해냈다. 더불어 실제 사용하는 이메일을 검증하기 위한 이메일 인증 서비스 및 도움이 필요한 사용자를 위한 도움말 서비스와 기존에 협업에서 사용하던 어려운 용어들을 실생활 용어로 바꾸는 등의 작업을 수행해내었다.</p>
4. 팀원 역할 분담	<ul style="list-style-type: none"> <li>- 이은채 (팀장) : 디자인 설계 및 프론트엔드 개발</li> <li>- 김나연 (팀원) : 데이터베이스 구축 및 백엔드 서버 개발</li> </ul>



5. 결과물 사진 또는 동작 화면



	예산배정금액	지출금액	잔액	비고
	2,400,000원	2,387,437원	12,563원	
6. 비용분석		항목	세부항목	소요비용
재료비		chatgpt : 474,593원 Claude : 326,256원 Cursor: 58,950원 Kling : 98,538원 캐럿 : 29,000원		987,437원
회의비		120,000원 * 5회		600,000원
전문가활용비		200,000원 * 4회		800,000원
		합 계		2,387,437원
7. 기타				
8. 참고문헌				

**9. 작품사진**

