



프로젝트 계획서



92113633 백이랑



프로젝트 소개 절차



01

프로젝트 주제 및 선정 이유

02

2. 프로젝트 구현 내용,)

03

3. 활용 라이브러리 및 프로그램 워크 (개발 환경)

04

4. 프로젝트 진행 프로세스 (WBS)

05

5. 프로젝트 목표

06

6. 기대효과

01

프로젝트 주제 및 선정 이유



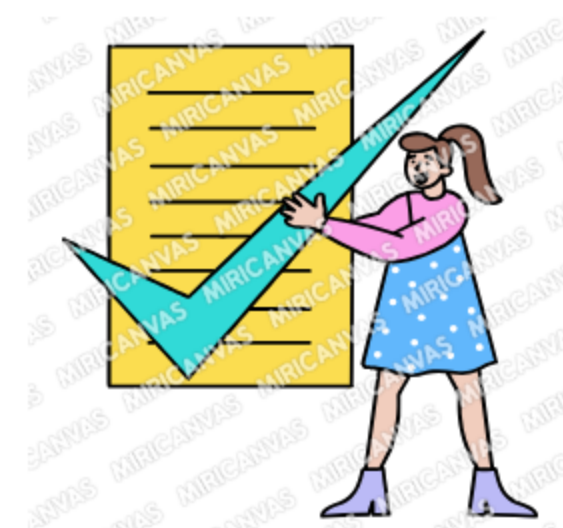
프로젝트 주제

간단한 방명록 웹 애플리케이션 제작



선정 이유

방명록은 사용자가 메시지를 남기고 확인할 수 있는 간단한 기능으로, 백엔드와 프론트엔드의 기본적인 상호작용을 학습하기에 판단되었기 때문에 선정해봤다.



02

2. 프로젝트 구현 내용,)



1. 구현내용



메시지 입력: 사용자가 이름과 메시지를 입력하여 방명록에 저장.

메시지 목록 표시: 저장된 메시지를 리스트 형태로 표시(이름, 메시지, 작성 시간).

디자인: 깔끔하고 간단한 UI로 메시지 입력과 목록을 보기 좋게 구성.

2. 구조



프론트엔드: HTML로 구조 작성, CSS로 스타일링.

백엔드: Python Flask로 서버 로직 및 API 구현.

데이터베이스: SQLite에 메시지 정보(이름, 메시지, 작성 시간) 저장.

03

3. 활용 라이브러리 및 프로그램 워크



front-end



HTML: 페이지 구조.
CSS: 스타일링 및 레이아웃.

back-end



Python Flask: 웹 서버 및 라우팅.
SQLite: 메시지 데이터 저장.

development environment



IDE: Visual Studio Code.
패키지 관리: pip.
브라우저: Chrome(테스트 및 디버깅).



03

3. 활용 라이브러리 및 프로그램 워크 (개발 환경)



4월	5월	6월	7월	8월	9월	10월	11월	12월
프로젝트 구성								
프로그램 공부 및 프로젝트 개발								
최종 점검 및 검토								
완성								

04

4. 프로젝트 진행 프로세스 (WBS)



• 1.요구사항 분석 및 설계

방명록에 포함할 정보 정리(이름, 메시지, 작성 시간).
데이터베이스 스키마 설계(테이블: ID, 이름, 메시지, 작성 시간).
API 엔드포인트 정의(/add, /view).

• 2.백엔드 개발

Flask 서버 설정 및 SQLite 연결.
API 구현:
/add: 새 메시지 저장.
/view: 저장된 메시지 조회.



• 3.프론트엔드 개발

index.html: 메시지 입력 폼과 목록 표시 UI 설계.
CSS로 깔끔한 스타일링 적용.

• 4.통합 및 테스트

프론트엔드와 백엔드 연결.
메시지 입력 → 조회 시나리오 테스트.

05

프로젝트 목표



기능 구현: 메시지 입력 및 목록 조회 기능이 정상 작동.

데이터 관리: SQLite를 통해 메시지를 영구적으로 저장 및 조회.

기술 학습: Flask와 SQLite를 활용한 백엔드 개발 익히기.

완성도: 오류 없이 동작하는 방명록 웹 애플리케이션 완성.

06

기대효과



실용성

간단한 방명록을 통해 방문자가 메시지를 남길 수 있음



학습 효과

프론트엔드와 백엔드의 기본적인 통합 경험.



확장 가능성

메시지 삭제 기능, 사용자 인증, 스타일링 개선 가능.



자신감 향상

동적 웹 애플리케이션 제작으로 개발 역량 강화

감사합니다.