



5일차: Flask 미니 프로젝트 ✨

4일차 복습

WebSocket이 필요한 이유

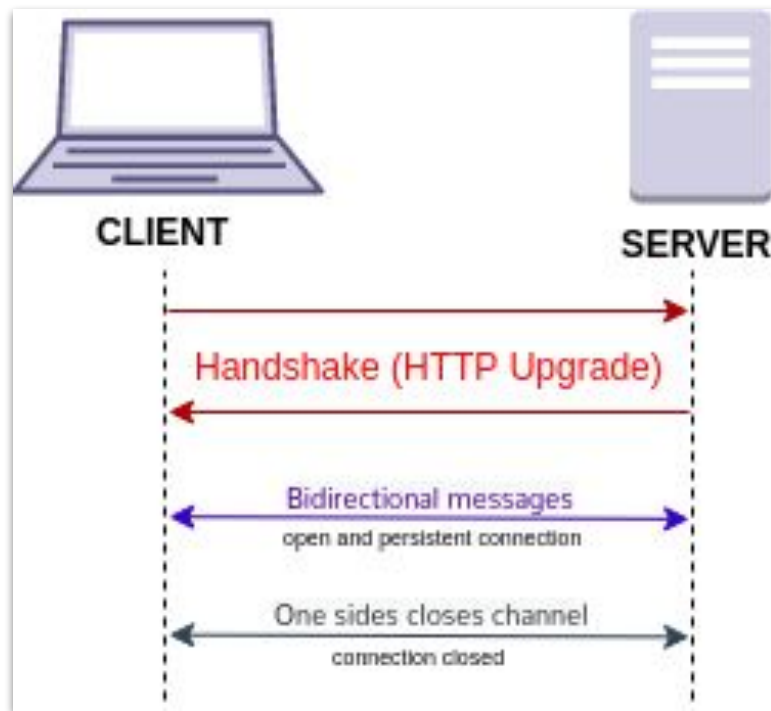
= 기존 HTTP 방식의 문제점

- 실시간성 부족
 - HTTP 요청-응답 방식은 클라이언트가 요청해야만 서버가 응답 가능
 - 실시간 알림, 채팅 등에는 비효율적
- 폴링(Polling)과 긴 연결(Long Polling)의 한계
 - 폴링: 클라이언트가 일정 시간마다 서버에 요청 (과부하 문제)
 - 긴 연결: 클라이언트가 연결을 유지하며 서버 응답을 기다림 (비효율적)

HTTP vs WebSocket 비교

항목	HTTP	WebSocket
연결	요청마다 새로	한 번 연결 유지
통신	단방향 (Client -> Server)	양방향
사용 예시	게시판, 블로그	채팅, 알림, 실시간 주식

WebSocket 프로토콜 구조



프로젝트

목표

- Flask로 CRUD 기능 전체 구현
- SQLite + SQLAlchemy를 활용한 DB 연동 복습
- 템플릿(Jinja2)으로 UI를 만들고, 별점 평균을 계산해 출력
- 지금까지 배운 내용을 종합해 “작동하는 서비스” 완성하기



프로젝트 개요

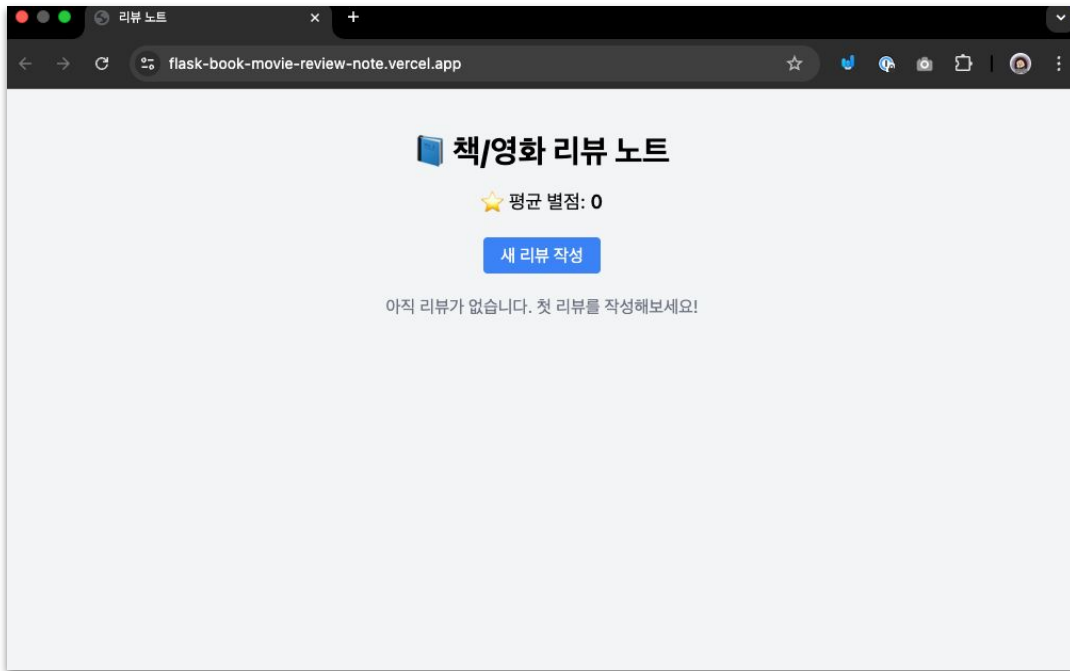
- 주제: 책/영화 리뷰 노트





프로젝트 개요: 책/영화 리뷰 노트

- 데모



<https://flask-book-movie-review-note.vercel.app/>



프로젝트 개요: 책/영화 리뷰 노트

- 주요 기능

- 리뷰 작성 (Create): 제목, 리뷰 내용, 별점 (1~5) 입력
- 리뷰 목록 보기 (Read)
 - 전체 리뷰 리스트 출력
 - ★ 평균 별점 표시
- 리뷰 수정 (Update): 기존 리뷰 내용 / 별점 수정
- 리뷰 삭제 (Delete): 리뷰 삭제 버튼



프로젝트 개요: 책/영화 리뷰 노트

- 최종 화면 예시
 - /: 리뷰 목록 + 평균 별점
 - /new: 리뷰 작성 폼
 - /edit/<id>: 리뷰 수정
 - /delete/<id>: 리뷰 삭제



프로젝트 개요: 책/영화 리뷰 노트

- 추천 진행 순서
 - 하나의 python 파일에서 in-memory로 구현하기
 - in-memory를 sqlite로 변경하기
 - 프로젝트 구조화하기 -> 다한 사람은 [구글폼](#) 작성
 - (선택) Vercel에 직접 배포하기: 5일차 README.md 참고

오전 11시: 3명 발표

오후 1시 30분: 코드리뷰 + 배포

QnA