

프로젝트 리뷰 보고서

작성자: DeepSeek API
작성일자: 2025-01-21

1. 리뷰 데이터 요약

PR ID	제목	평균 등급	작성일자
6	Test PR Review 1	B	2025-01-19
12	Revert "[feat/#57] 보고서 제작에 DEEPSEEK API 연동 구현"	B	2025-01-20
20	Revert "[feat/#57] 보고서 제작에 DEEPSEEK API 연동 구현"	B	2025-01-20
26	Revert "[feat/#57] 보고서 제작에 DEEPSEEK API 연동 구현"	A	2025-01-21
27	Revert "[feat/#57] 보고서 제작에 DEEPSEEK API 연동 구현"	A	2025-01-21

2. 분석 내용

2-1. 리뷰 결과 통계

- **분석된 PR 수**: 5
- **Clean 모드**: 1개의 리뷰
- **Optimize 모드**: 0개의 리뷰
- **Study 모드**: 0개의 리뷰
- **newbie 모드**: 0개의 리뷰
- **basic 모드**: 0개의 리뷰

2-2. 주요 취약점 및 개선 우선순위

취약한 유형 통계 및 개선 방향:

1. **취약점 유형 문제점**:

- **코드 가독성**: 변수명이 명확하지 않거나, 코드 구조가 복잡하여 이해하기 어려운 경우가 많았습니다.
- **중복 코드**: 동일한 로직이 여러 곳에서 반복되어 유지보수가 어렵습니다.

- ****에러 처리 미흡****: 예외 상황에 대한 처리가 부족하여 프로그램이 비정상적으로 종료될 가능성이 있습니다.

2. ****개선 방향****:

- ****코드 가독성****: 변수명을 명확하게 지정하고, 함수를 적절히 분리하여 코드의 가독성을 높이세요.

- ****중복 코드****: 공통 로직을 함수로 추출하여 재사용성을 높이세요.

- ****에러 처리****: 모든 예외 상황을 고려하여 적절한 에러 처리를 추가하세요.

3. ****안좋은 예시와 좋은 예시****:

- ****안좋은 예시****:

```
```python
def calculate(a, b):
 result = a + b
 return result
```
```

- 변수명이 명확하지 않아 코드의 의도를 파악하기 어렵습니다.

- ****좋은 예시****:

```
```python
def add_two_numbers(first_number, second_number):
 sum_result = first_number + second_number
 return sum_result
```
```

- 변수명이 명확하여 코드의 의도를 쉽게 파악할 수 있습니다.

****2-3. 개인화된 피드백 및 권장사항****

****사용자 맞춤 개선 방향****:

- ****가장 낮은 점수를 받은 평가 기준****: 코드 가독성

- ****개선 방안****:

- 변수명을 명확하게 지정하고, 함수의 역할을 명확히 정의하세요.
- 코드를 작은 단위로 분리하여 각 함수가 하나의 역할만 수행하도록 하세요.
- 주석을 적절히 사용하여 코드의 의도를 명확히 전달하세요.

****2-4. 종합 결론****

- **총평:**

- 프로젝트의 전체적 성향은 기본적인 기능 구현에 초점이 맞춰져 있으나, 코드의 가독성과 유지보수성에서 개선 여지가 큼니다. 특히, 코드 가독성과 중복 코드 문제를 해결한다면 프로젝트의 품질이 크게 향상될 것입니다.

- **강점:**

1. 기본적인 기능 구현이 잘 되어 있습니다.
2. 코드의 논리적 구조가 명확합니다.
3. 프로젝트의 목표를 달성하기 위한 핵심 로직이 잘 구현되어 있습니다.

- **약점:**

1. 코드 가독성이 낮아 이해하기 어렵습니다.
2. 중복 코드가 많아 유지보수가 어렵습니다.
3. 에러 처리가 미흡하여 프로그램의 안정성이 떨어집니다.

- **향후 권장 사항:**

- ****Clean 모드****를 사용하여 코드의 가독성과 유지보수성을 높이는 데 집중하세요.
- 코드 리팩토링을 통해 중복 코드를 제거하고, 함수의 역할을 명확히 분리하세요.
- 예외 처리를 강화하여 프로그램의 안정성을 높이세요.

****첨부 자료****

- **추천 학습 자료:**

- [Clean Code by Robert C. Martin](<https://www.amazon.com/Clean-Code-Handbook-Software-Craftsmanship/dp/0132350882>)

- [Refactoring: Improving the Design of Existing Code by Martin Fowler](<https://www.amazon.com/Refactoring-Improving-Design-Existing-Code/dp/0201485672>)

- **관련 예시 코드**:

```
```python
def calculate_discount(price, discount_rate):
 if discount_rate < 0 or discount_rate > 1:
 raise ValueError("Discount rate must be between 0 and 1")
 return price * (1 - discount_rate)
```
```

- 이 코드는 가독성이 높고, 에러 처리가 잘 되어 있습니다.
