

프로젝트 리뷰 보고서

작성자: DeepSeek API
작성일자: 2025-01-22

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, 코드의 의도를 파악하기 어려운 경우가 많음.
- **중복 코드**: 동일한 로직이 여러 곳에서 반복되어 유지보수가 어려움.

- ****에러 처리 미흡****: 예외 상황에 대한 처리가 충분히 이루어지지 않아 안정성이 떨어짐.

2. ****개선 방향****:

- ****코드 가독성****: 변수명을 명확하게 지정하고, 코드의 의도를 주석으로 설명.

- ****중복 코드****: 공통 로직을 함수로 분리하여 재사용성을 높임.

- ****에러 처리****: 예외 상황에 대한 처리를 강화하여 안정성을 높임.

3. ****안좋은 예시와 좋은 예시****:

- ****안좋은 예시****:

```
python
```

```
a = 10
```

```
b = 20
```

```
c = a + b
```

```
...
```

- ****좋은 예시****:

```
python
```

```
first_number = 10
```

```
second_number = 20
```

```
sum_of_numbers = first_number + second_number
```

```
...
```

```
---
```

****2-3. 개인화된 피드백 및 권장사항****

****사용자 맞춤 개선 방향****:

- ****가장 낮은 점수를 받은 평가 기준****: 코드 가독성

- ****개선 방안****:

- 변수명을 명확하게 지정하여 코드의 의도를 쉽게 파악할 수 있도록 함.

- 함수와 클래스의 역할을 명확히 정의하고, 주석을 통해 설명을 추가.

- 코드 리뷰를 통해 동료 개발자와의 협업을 통해 가독성을 높이는 방법을 학습.

****2-4. 종합 결론****

- **총평:**

- 프로젝트의 전체적 성향은 기본적인 기능 구현에 중점을 두고 있으며, 코드의 가독성과 중복 코드 문제에서 개선 여지가 큼니다. 특히, 코드 가독성을 높이는 방향으로 개선한다면 프로젝트의 유지보수성이 크게 향상될 것입니다.

- **강점:**

1. 기본적인 기능 구현이 잘 되어 있습니다.
2. 코드의 구조가 간결하고 이해하기 쉽습니다.
3. 프로젝트의 목표를 명확히 이해하고 있습니다.

- **약점:**

1. 코드 가독성이 떨어져 유지보수가 어렵습니다.
2. 중복 코드가 많아 코드의 재사용성이 낮습니다.
3. 에러 처리에 대한 고려가 부족하여 안정성이 떨어집니다.

- **향후 권장 사항:**

- ****Clean 모드****를 사용하며 코드의 가독성과 유지보수성을 높이는 역량을 키우는 것을 추천합니다.
- 코드 리뷰를 통해 동료 개발자와의 협업을 통해 코드 품질을 높이는 방법을 학습하세요.

****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_sum(numbers):
```

```
return sum(numbers)
```

```
def calculate_average(numbers):
```

```
 return sum(numbers) / len(numbers) if numbers else 0
```

```
'''
```

```
'''
```