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

```markdown
---

**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**:
- **취약점 유형 문제점**:

1. **코드 가독성**: 변수명이 명확하지 않거나, 코드 구조가 복잡하여 이해하기 어려움.
```

2. **중복 코드**: 동일한 로직이 여러 곳에서 반복되어 유지보수가 어려움.

3. **에러 처리 미흡**: 예외 상황에 대한 처리가 부족하여 프로그램이 비정상적으로 종료될 가능성이 높음.

- **개선 방향**:

1. **코드 가독성**: 변수명을 명확하게 지정하고, 함수를 적절히 분리하여 코드의 의도를 명확히 표현.

2. **중복 코드**: 공통 로직을 함수로 추출하여 재사용성을 높임.

3. **에러 처리**: 예외 상황을 고려하여 적절한 에러 핸들링을 추가.

- **안좋은 예시와 좋은 예시**:

- **안좋은 예시**:

```python
def process_data(data):
for item in data:
if item['status'] == 'active':
# 복잡한 로직
pass
```

- **좋은 예시**:

```python
def is_active(item):
return item['status'] == 'active'

def process_data(data):
for item in data:
if is_active(item):
# 명확한 로직
pass
```

2-3. 개인화된 피드백 및 권장사항

사용자 맞춤 개선 방향:
- **가장 낮은 점수를 받은 평가 기준**: 코드 가독성
- **개선 방안**:

1. **변수명 개선**: 변수명을 명확하고 직관적으로 변경하여 코드의 의도를 쉽게 이해할 수 있도록 함.

2. **함수 분리**: 복잡한 로직을 작은 함수로 분리하여 각 함수가 하나의 역할만 수행하도록 함.

3. **주석 추가**: 코드의 중요한 부분에 주석을 추가하여 다른 개발자가 쉽게 이해할 수 있도록 함.

2-4. 종합 결론

- **총평**:

- 프로젝트의 전체적 성향 및 평균 등급을 출력하고, 코드 가독성 부분에서 개선 여지가 가장 큽니다. 이를 통해 코드의 유지보수성을 높일 수 있습니다.

- **강점**:

1. **기능 구현**: 주요 기능들이 잘 구현되어 있습니다.

2. **모듈화**: 일부 코드가 모듈화되어 있어 재사용성이 높습니다.

3. **성능**: 기본적인 성능 최적화가 잘 이루어져 있습니다.

- **약점**:

1. **코드 가독성**: 변수명과 함수명이 명확하지 않아 코드 이해가 어렵습니다.

2. **중복 코드**: 동일한 로직이 여러 곳에서 반복되어 있습니다.

3. **에러 처리**: 예외 상황에 대한 처리가 미흡합니다.

- **향후 권장 사항**:

- **Clean 모드**를 사용하며 코드의 가독성과 유지보수성을 높이는 데 집중하세요.

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

# 좋은 예시: 명확한 변수명과 함수 분리

def calculate_discount(price, discount_rate):

return price * (1 - discount_rate)

def apply_discount(items, discount_rate):

for item in items:

item['price'] = calculate_discount(item['price'], discount_rate)

```

```
```