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**:

1. **취약점 유형 문제점**: 코드 가독성 부족
```

- **개선 방향**: 변수명과 함수명을 직관적이고 명확하게 변경하여 코드의 의도를 쉽게 파악할 수 있도록 개선.

- **안좋은 예시**: `int a = 10;`

- **좋은 예시**: `int userAge = 10;`

2. **취약점 유형 문제점**: 중복 코드

- **개선 방향**: 중복된 코드 블록을 함수로 추출하여 재사용성을 높이고 유지보수를 용이하게 함.

- **안좋은 예시**:

```python
print("Hello, World!")
print("Hello, World!")
```

- **좋은 예시**:

```python
def greet():
print("Hello, World!")

greet()
greet()
```

3. **취약점 유형 문제점**: 예외 처리 미흡

- **개선 방향**: 예외 상황을 고려하여 적절한 예외 처리를 추가하여 프로그램의 안정성을 높임.

- **안좋은 예시**:

```python
result = 10 / 0
```

- **좋은 예시**:

```python
```

try:

result = 10 / 0

except ZeroDivisionError:

print("Cannot divide by zero")

```

---

**2-3. 개인화된 피드백 및 권장사항**

**사용자 맞춤 개선 방향**:

- **가장 낮은 점수를 받은 평가 기준**: 코드 가독성

- **개선 방안**: 변수명과 함수명을 더 직관적이고 명확하게 변경하여 코드의 의도를 쉽게 파악할 수 있도록 개선하세요. 예를 들어, `a`와 같은 모호한 변수명 대신 `userAge`와 같이 명확한 이름을 사용하세요.

- **보편적 개선안**:

- **코드 리팩토링**: 중복 코드를 함수로 추출하여 재사용성을 높이고 유지보수를 용이하게 하세요.

- **예외 처리 강화**: 예외 상황을 고려하여 적절한 예외 처리를 추가하여 프로그램의 안정성을 높이세요.

---

**2-4. 종합 결론**

- **총평**:

- 프로젝트의 전체적 성향 및 평균 등급을 출력하고, 코드 가독성 부분에서 개선 여지가 가장 큽니다. 코드의 의도를 명확히 표현하고, 중복 코드를 줄이며, 예외 처리를 강화하면 더욱 견고한 코드를 작성할 수 있습니다.

- **강점**:

1. 코드의 기본 구조가 잘 잡혀 있습니다.

2. 기능 구현이 명확하고 목적에 부합합니다.

3. 코드의 실행 속도가 빠릅니다.

- **약점**:

1. 코드 가독성이 떨어집니다.

2. 중복 코드가 많아 유지보수가 어렵습니다.
```

3. 예외 처리가 미흡하여 프로그램의 안정성이 떨어집니다.

- **향후 권장 사항**:

- **Clean 모드**를 사용하며 코드의 가독성과 유지보수성을 높이는 역량을 키우세요. 변수명과 함수명을 명확히 하고, 중복 코드를 줄이며, 예외 처리를 강화하는 연습을 하세요.

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: 가독성 높은 코드

def calculate_user_age(birth_year):

current_year = 2023

return current_year - birth_year

user_age = calculate_user_age(1990)

print(f"User's age is {user_age}")

```

```python

# 좋은 예시: 예외 처리 강화

def divide_numbers(a, b):

try:

return a / b

except ZeroDivisionError:

return "Cannot divide by zero"
```

```
result = divide_numbers(10, 0)

print(result)
```

```
```