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**: 변수 및 함수 네이밍을 직관적으로 변경하고, 주석을 추가하여 코드의 의도를 명확히 표현하세요.

- **안좋은 예시**: `int a = 10;`

- **좋은 예시**: `int userAge = 10;`

2. **취약점 유형 문제점**: 중복 코드

- **개선 방향**: 중복된 코드 블록을 함수로 추출하여 재사용성을 높이세요.

- **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 try-catch 블록을 추가하세요.

- **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 기준은 **코드 가독성**입니다. 이를 개선하기 위해 다음과 같은 보편적 개선안을 제안합니다:

1. **변수 및 함수 네이밍**: 변수와 함수의 이름은 그 용도를 명확히 나타내야 합니다. 예를 들어, `a` 대신 `userAge`와 같이 의미 있는 이름을 사용하세요.

2. **주석 추가**: 복잡한 로직이나 중요한 부분에는 주석을 추가하여 코드의 의도를 설명하세요.

3. **코드 포맷팅**: 일관된 들여쓰기와 줄바꿈을 사용하여 코드를 깔끔하게 정리하세요.

---

**2-4. 종합 결론**

- **총평**:

- 프로젝트의 전체적 성향은 기본적인 기능 구현에 초점이 맞춰져 있으며, 평균 등급은 중간 수준입니다. 특히 **코드 가독성**과 **중복 코드** 부분에서 개선 여지가 가장 큽니다. 이러한 부분을 보완한다면 코드의 품질이 크게 향상될 것입니다.

- **강점**:

1. 기본적인 기능 구현이 잘 되어 있습니다.

2. 코드의 구조가 비교적 단순하여 이해하기 쉽습니다.

3. 필요한 기능을 빠르게 구현할 수 있는 능력이 있습니다.

- **약점**:

1. 코드 가독성이 낮아 이해하기 어렵습니다.

2. 중복 코드가 많아 유지보수가 어렵습니다.

3. 예외 처리가 미흡하여 안정성이 떨어집니다.
```

- **향후 권장 사항**:

- **Clean 모드**를 사용하며 코드의 가독성과 유지보수성을 높이는 역량을 키우는 것이 중요합니다. 이를 통해 더 나은 코드 품질을 유지할 수 있습니다.

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 age: {user_age}")

```

```
```