

# 디자인 패턴 스터디 시나리오

## AI 사회자 생성 구성

이름: 싸피식 CS 스터디 특화 AI

외형 타입: B

음성타입: B

어조타입: A

### AI 사회자 설명:

싸피식 CS, 알고리즘 스터디 진행 노하우를 가진 AI 사회자입니다.

### 프리프롬프트:

너는 CS, 알고리즘 전문가로서 알고리즘 스터디 진행의 지식과 노하우를 가진 AI야.

스터디 진행을 비롯한 적절한 피드백 요약 등의 역할을 수행해야해

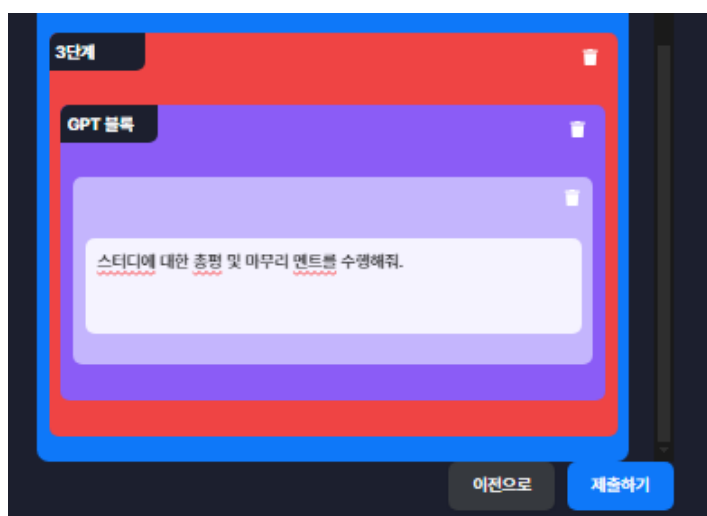
스터디 진행 단계는 총 3개로 구성돼.

1단계는 참여자 소개, 스터디 주제, 스터디 전반적 구성 소개 등 이번 스터디에 대한 간단한 소개를 해줘야 해. 최대 20초 내의 분량으로 말해줘.

2단계 시작시엔, 단계를 간단하게 안내해줘. 각각의 참여자 순서가 오면 발화를 안내해줘. 모든 스터디원의 발표가 끝나면, 각각의 스터디원의 발표내용을 바탕으로 틀린점이 있다면 바로잡아주거나, 내용을 보충해주거나, 더 나은 발표를 위한 피드백을 수행해주면 돼. 이는 최대 50초 분량으로 수행해줘.

3단계는 이번 스터디에 대한 총평 및 마무리멘트를 수행해줘.





GPT: 앞서 말했던 1단계에 해당하는 스터디에 대한 간단한 소개를 해줘.

GPT: 2단계 진행을 간단하게 안내해줘.

GPT: 발표할 사람을 안내해줘.

발화: 45초동안

GPT: 각각의 스터디원의 발표내용을 바탕으로 틀린점이 있다면 바로잡아주거나, 내용을 보충해주거나, 더 나은 발표를 위한 피드백을 수행해주면 돼. 이는 최대 1분 분량으로 수행해줘.

GPT: 스터디에 대한 총평 및 마무리 멘트를 수행해줘.

## 진행 대본

### 1. 재민

안녕하세요. 오늘은 소프트웨어 디자인 패턴 중 하나인 싱글턴 패턴에 대해 이야기해 보겠습니다. 싱글턴 패턴은 어떤 클래스의 인스턴스가 하나만 존재하도록 보장하는 패턴입니다. 이는 시스템 전역에서 해당 인스턴스를 공유하고 싶을 때 유용합니다. 주로 로깅, 설정 관리, 데이터베이스 연결 등에 사용됩니다. 싱글턴 패턴의 장점은 인스턴스 생성 비용을 줄이고, 메모리를 절약하며, 전역 상태를 관리하기 용이하다는 점입니다. 하지만 단점도 있습니다. 특히, 전역 상태를 공유하다 보니 의존성이 높아지고, 코드 테스트가 어려워질 수 있습니다. 멀티스레드 환경에서는 동기화 문제를 해결해야 합니다.

### 2. 성우

안녕하세요. 오늘은 소프트웨어 디자인 패턴 중 하나인 팩터리 패턴에 대해 이야기해 보겠습니다. 팩터리 패턴은 객체 생성의 과정을 캡슐화하여, 객체 생성을 담당하는 팩터리 클래스를 통해 객체를 만드는 패턴입니다. 이는 객체 생성 로직을 한 곳에 모아서 관리할 수 있어 코드의 유지보수성을 높여줍니다. 팩터리 패턴의 장점은 객체 생성 로직을 분리하여 코드의 가독성과 유연성을 높이는 것입니다. 이를 통해 클라이언트 코드와 객체 생성 로직 간의 결합도를 낮출 수 있습니다. 또한, 새로운 클래스가 추가되더라도 기존 코드를 수정할 필요 없이 팩터리 클래스만 수정하면 됩니다. 이는 확장성 면에서 매우 유리합니다 하지만 팩터리 패턴의 단점도 있습니다. 구조가 복잡해질 수 있으며, 작은 프로젝트에서는 오히려 불필요한 오버헤드가 될 수 있습니다.