

# 프로그래밍 학습을 위한 LLM 기반 코드 리뷰 학습 환경 개발

## Designing LLM-based Code Reviewing Learning Environment for Programming Education

최승윤<sup>†</sup> · 이동건<sup>††</sup> · 김준구<sup>††</sup> · 장연주<sup>†††</sup> · 김현철<sup>††††</sup>

Seongyune Choi<sup>†</sup> · Dong-geon Lee<sup>††</sup> · Jungu Kim<sup>††</sup> · Yeonju Jang<sup>†††</sup> · Hyeoncheol Kim<sup>††††</sup>

### 요 약

프로그래밍 학습에서 코드 리뷰(Code review)는 효과적인 방법이지만, 그러한 리뷰를 받을 수 있는 환경은 매우 제한적이다. 이에 대한 해결책으로 AI 기반의 코드 리뷰 시스템에 관한 연구가 진행됐으나, 기존에 개발된 시스템들은 학습자의 접근 방식이 다양하여 학습자 맞춤형 피드백 제공이 어려웠고, 또한 코드와 관련된 질문에 대한 응답도 제공하기 어려웠다. 거대 언어모델(Large Language Model, LLM)은 이러한 코드 리뷰 시스템 개발에 새로운 접근법이 될 수 있다. 하지만, 프롬프트의 설계에 따라 성과와 활용 목적이 바뀌는 LLM의 특성을 고려하였을 때 효과적인 프롬프트 구축에 관한 연구는 필수적이나, 관련 연구는 아직 부족한 실정이다. 이에 본 연구에서는 프롬프트 프로토콜을 구축하고 LLM을 활용한 대화형 코드 리뷰 시스템을 개발하고자 하였다. 본 연구는 선행 연구 분석을 통해 설계 원리를 도출하고 이를 기반으로 다양한 개발 환경을 사용하여 웹 시스템을 구축하였다. 개발된 시스템은 총 12인이 포함된 사용자 평가를 통해 개선되었으며, 개선된 시스템에서 제공하는 피드백 및 응답은 '적절성', '구체성', '명확성', '유용성' 네 가지 평가 기준에 따라 평가되었다. 개발된 시스템은 학습자가 코드를 스스로 개선할 수 있도록 코드에 대한 피드백을 제공하며, 관련 질문을 할 수 있는 환경을 제공함으로써 학교 현장에서의 수업 도구로 사용 가능할 뿐만 아니라, 스스로 학습하는 데 유용하게 활용될 수 있다. 또한, 본 연구에서 구축된 프롬프트 프로토콜은 LLM을 SW 교육에 적용할 때의 기반이 될 수 있을 것으로 기대된다.

**주제어:** 거대 언어모델, 프로그래밍 교육, 코드 리뷰

### ABSTRACT

Code review is an effective method in programming education; however, opportunities for receiving such reviews are often limited. To address this issue, researchers have explored AI-based code review systems, but existing systems face challenges in providing customized feedback to learners with diverse approaches and responding to their questions. Large Language Models (LLMs) offer a promising new approach to developing code review systems. Given that LLMs' performance and purposes vary based on prompt design, research on effective prompt construction is essential. However, the knowledge in designing such prompt protocols remains limited. In this study, we aimed to develop a conversational code review system using LLMs by constructing a prompt protocol. We derived design principles through the analysis of prior research and built a web-based system based on these principles. The developed system underwent iterative improvement through user evaluations involving 12 participants, and the feedback and responses provided by the system were assessed. The system supports learners by not only offering feedback on their code but also addressing their questions. This tool can be used in various educational settings, such as K-12 or higher education, and can be effectively utilized for self-study. Furthermore, this study is expected to lay the foundation for prompt engineering when applying LLMs to software education.

**Keywords:** LLM, programming education, code review

<sup>†</sup>정 회 원: 고려대학교 일반대학원 컴퓨터학과 박사수료

<sup>††</sup>정 회 원: 고려대학교 일반대학원 컴퓨터학과 석사과정

<sup>†††</sup>정 회 원: 고려대학교 정보창의교육연구소 연구원

<sup>††††</sup>중신회원: 고려대학교 컴퓨터학과 교수(교신저자)

논문투고: 2023년 05월 11일, 심사완료: 2023년 07월 24일, 게재확정: 2023년 07월 26일

## 1. 서론

최근 급격한 ICT의 발달은 디지털 전환(Digital transformation)을 야기하였고, 이에 따라 디지털 역량은 가장 주요한 역량 중 하나로 여겨진다[1]. 15개 국가에서 18,000여 명을 대상으로 연구한 매킨지의 보고서에서는 디지털 역량이 인지 능력, 대인관계, 자기 리더십과 더불어 가장 주요한 직업 역량으로 자리매김하였음을 확인하였다[2]. 이처럼, 일상적인 업무에 ICT 도구를 다루는 능력과 같이 기본적인 역량은 필수사항이 되었고, 점점 더 많은 업무에서 데이터 처리, 분석, SW 개발 등과 같은 상대적으로 높은 수준의 디지털 역량도 점점 더 요구되고 있다[3-4].

이러한 디지털 역량 교육의 일환으로 프로그래밍 교육에 대한 수요와 그 필요성은 점점 더 증대되어 오고 있으며, 효과적인 교육을 위한 연구들과 정책들이 뒷받침되고 있다[5-7]. 선행 연구에 따르면, 프로그래밍 학습이 효과적으로 이루어지기 위해서는 배운 개념을 적용하여 문제를 해결하는 활동이 필요하며, 배운 개념을 문제에 적용하지 않거나 단순히 사실을 암기하는 등의 활동만으로는 학습이 제대로 이루어지지 않는다[8]. 그런데 프로그래밍은 그 특성상 학습 시 다양한 풀이가 가능하다. 학습자마다 작성한 코드가 서로 다르고 접근하는 방식이 다르므로 학습자 맞춤형 지원이 필요하며, 동료 학습자나 전문가 등에게 코드를 리뷰 받는 코드 리뷰(Code review)가 효과적인 접근이 될 수 있다[9].

하지만, 코드 리뷰를 받을 수 있는 환경은 매우 제한적이기 때문에 프로그래밍 학습의 진입장벽이 높다는 문제가 있다. 특히, 학교 현장에서 학습자는 대부분 교사에게 의존하여 코드의 리뷰를 받으며 학습하고 있다. 하지만 많은 학생을 관리해야 하는 학교의 특성상 교사가 개별적으로 지원하기란 매우 어려운 실정이다. 또한, 스스로 학습하는 경우에도 마찬가지로 도움을 요청할 만한 사람이 마땅치 않다[10].

이를 보완하기 위하여 AI 기반의 자동 피드백 제공 시스템에 관한 연구가 진행됐다[11]. 하지만, 이러한 시스템들은 보통 논리 오류와 관련한 피드백은 제공되기 어려웠고, 코드의 변화되는 양상을 고려한 피드백을 제공해주기 어려운 한계가 있었다. 또한, 코드와 관련한 질문에 대한 응답을 제공해주기 어려운 한계가 있었다.

거대 언어모델(Large language model, LLM)은 이러한 한계를 보완하는 효과적인 접근법일 수 있다. LLM

은 대량의 텍스트 데이터로 훈련되어 인간과 유사한 텍스트를 생성하고, 질문에 답변하며, 언어 관련 작업을 높은 정확도로 수행할 수 있다[12]. LLM은 텍스트 뿐 아니라 프로그래밍 코드의 분석 및 생성에도 좋은 성능을 보여주고 있어서, 학습용 코드 리뷰 시스템 개발에 유용한 접근방법이 될 수 있다.

하지만, LLM을 기반으로 한 프로그래밍 학습용 코드 리뷰 시스템에 관한 연구는 매우 미흡하며 이에 따라 LLM을 활용한 교육용 코드 리뷰 시스템에서 프롬프트의 설계는 어떻게 구성되어야 하는지도 잘 연구되지 않았다. 특히, 프롬프트의 설계에 따라 그 성과와 활용 목적이 바뀌는 LLM의 특성에 근거하였을 때 효과적인 프롬프트 구축에 관한 연구는 필수적이라고 볼 수 있다[13].

따라서 본 연구는 프로그래밍 학습을 위한 코드 리뷰 시스템을 개발하고자 한다. 이때 개발에 있어 LLM을 활용할 때의 적절한 프롬프트 디자인과 프로토콜을 구축하고 이를 바탕으로 통합적 프로그래밍 학습 환경을 개발하고자 한다.

## 2. 이론적 배경

### 2.1 코드 리뷰 시스템

코드 리뷰는 작성한 코드의 버그를 찾고 피드백을 제공하여 코드의 품질을 향상하는 것으로 소프트웨어 개발에 있어 중요한 역할을 하고 있다[14]. 프로그래밍 학습에 있어서도 코드 리뷰를 통해 학습자의 코드를 검토하고 그에 대한 피드백을 제공함으로써 학습자의 프로그래밍 학습을 효과적으로 지원할 수 있다[15]. 하지만, 이와 같은 코드의 리뷰는 교육 자원이 제한적인 상황에서 개별적인 수준에서 맞춤형으로 제공되기 어렵다.

이를 보완하기 위하여 학습자의 코드를 검토하고 자동으로 피드백을 줄 수 있는 시스템들이 제시되었다[11]. 이 시스템들은 기계학습[16], 탐색[17] 과 같은 AI 기법이 주로 사용되었고, 학습자의 코드를 기준 코드와의 유사성을 비교하여 피드백을 제공하여 학습자의 학습을 지원한다.

하지만, 이러한 접근은 단순히 오류를 지적하는 것으로 학습자가 교착 상태에서 해결책을 찾아 나가는 데 실질적인 도움이 되기 어렵고, 다양한 접근방법에 대한 맞춤형 학습 지원을 제공하는 데 한계가 있다. 또한, 코드와 관련된 학습자의 질문에 대한 응답을 제

공하는 기능은 제공하지 않아 학습자가 결과물을 개선하는 데 제한적이었다.

따라서 학습자가 코드에 대한 리뷰를 받을 때 학습자의 다양한 접근과 변화되는 코드의 양상을 고려한 코드 리뷰가 필요하며, 코드와 관련된 학습자의 질문에 대한 응답을 제공할 수 있는 시스템에 관한 연구가 필요하다.

## 2.2 LLM의 교육적 활용 가능성과 프롬프트 엔지니어링의 필요성

LLM은 방대한 양의 텍스트 데이터를 학습하여 만들어지며, 다양한 언어와 주제에 대한 이해를 바탕으로 텍스트 생성, 질문에 대한 답변, 감정 분석 등 다양한 작업을 수행할 수 있다[12]. 이러한 LLM은 기계 번역, 챗봇, 추천 시스템 등 다양한 분야에서 활용되며, 다양한 분야에 활용 가능성이 증가하고 있다. 최근 교육 분야에서 AI 기술을 접목하는 시도들이 점점 활발해지면서[18-20] LLM의 교육 분야에서의 활용도 화두가 되고 있다. 특히 LLM을 기반으로 한 생성적 AI 챗봇인 ChatGPT가 가져올 교육의 변화에 관심이 집중되고 있다[21].

이와 같은 LLM은 설정한 프롬프트에 따라서 생성되는 산출물의 결과가 상이하대[13]. 따라서, LLM을 교육적으로 의미있게 활용하기 위해서는 목적에 맞는 프롬프트 엔지니어링이 필요하다. LLM은 텍스트뿐만 아니라 프로그래밍 코드와 관련하여서도 좋은 성능을 보여주어 프로그래밍 수업에 효과적으로 적용될 수 있지만, 적절한 프롬프트 설정이 이루어지지 않으면 답을 바로 제시하거나 학습 내용과 관련 없는 피드백을 제공하는 등 교육적인 효과를 얻기 어렵다. 하지만, 하지만 이와 같은 LLM 기술을 실제로 어떻게 활용할 수 있을지에 관한 연구는 미흡한 상태이며, 따라서 학습용 코드 리뷰 시스템 개발에 유용한 프롬프트 프로토콜에 관한 연구가 필요하다.

## 3. 연구 방법 및 결과

### 3.1 연구 방법

본 연구는 선행 연구 분석, 설계 원리 도출, 설계 및 개발, 사용자 평가 및 개선, 피드백 및 응답의 검증 단계로 진행되었으며 Figure 1에 본 연구의 절차가 묘사되어 있다.

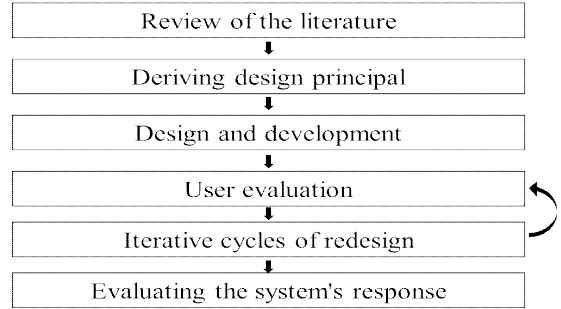


Figure 1. Research procedure

우선 선행 연구 분석 단계에서는 코드 리뷰 시스템과 LLM의 교육적 활용에 관한 연구를 분석하였고, 이를 토대로 설계 원리를 도출하였다. 다음으로 설계와 개발 단계에서는 개발 환경 구축, 모듈 기능 설계 및 구현, UI 설계 및 기능 통합을 절차에 따라 진행하였다. 다음으로, 개발된 시스템의 유효성을 총 세 차례의 사용자 평가를 통해 검증하였고, 반복적 재설계를 통하여 시스템을 개선하였다[22]. 마지막으로 시스템을 개선한 이후 본 시스템에서 제공되는 피드백 및 응답의 적절성을 검증하였다.

### 3.2 설계 원리 도출

LLM 기반 코드 리뷰 프로그래밍 학습 환경을 개발하기 위하여 문헌 연구를 통해 여섯 가지 설계 원리를 도출하였다.

첫째, ‘대화형 피드백 환경의 제공’이다. 대화형 AI 챗봇 시스템은 익숙한 인터페이스로 학습자들이 쉽게 사용할 수 있다는 장점이 있다[23]. 따라서, 코드 리뷰 기능은 챗봇과 채팅하는 형식으로 학습자의 학습을 지원하도록 설계되었다.

둘째, ‘일상적인 프로그래밍 환경의 제공’이다. 학습자들이 본 시스템에서 학습한 이후 전문적인 개발 환경을 접할 때 간극을 느끼지 않도록 비슷한 환경이 구축되어야 한다.

셋째, ‘정답 코드 생성 제한’이다. 정답 코드를 학습자에게 제시한다면 학습자가 스스로 시도할 기회가 제한되어 학습이 저해될 수 있다. 따라서 피드백 중 정답 코드가 포함되지 않도록 하며, 학습자가 정답을 요청하더라도 정답 생성이 제한되어야 한다.

넷째, ‘문제 풀이 확인 기능 제공’이다. 풀이본 문제에 대하여 정 오답 여부를 확인하는 과정은 학습에서 중요한 단계이므로 이를 고려한 설계가 필요하다.

다섯째, ‘논리 오류 관련 피드백과 코딩 습관 형성 관련 피드백 제공’이다. 논리 오류는 에러의 위치를 찾기가 쉽지 않고 수정하기도 어려운 일이다. 학습자는 이러한 로직에 대해 특히나 어려움을 갖고 있을 수 있으며, 이에 대한 피드백이 제공되면 학습에 더욱 유의미할 것이다. 또한, 적절한 변수 이름의 설정 등과 같은 코딩 습관이 가능한 피드백이 제공될 수 있도록 설계된다면 효과적인 코딩 습관 형성에 도움이 될 수 있다.

마지막으로, ‘단순하고 쉬운 조작이 가능한 구성’이다. 직관적이고 쉽게 조작할 수 있게 시스템이 설계된다면 학습자들은 별다른 노력 없이도 사용법을 익힐 수 있을 것이다.

### 3.3 설계 및 개발

위와 같이 도출한 개발원리를 기반으로 본 시스템을 구축하였다. 개발된 시스템은 백 엔드부터 프론트 엔드까지 통합적으로 구축되었으며 다양한 개발 환경이 사용되었다. Figure 2에 사용된 환경들이 묘사되어 있다.







Front-end	Web	React		ANTD	
	Server	Google Cloud Platform			
Back-end	Compiler module	Code check module	Peer review module		
	Python		GPT		
	Database	MySQL			

Figure 2. System configuration and environment

백 엔드 부분에는 본 시스템의 핵심 기능을 담당하는 세 가지의 모듈이 포함된다. 먼저, 컴파일러 모듈(Compiler module)은 학습자가 작성한 코드를 실행 실행시켜볼 수 있는 모듈로 학습자의 코드를 컴파일하고, 결과를 학습자에게 제공하는 일상적인 코드 환경을 제공한다. 본 연구에서는 프로그래밍 학습에서 많이 다루는 언어인 파이썬(Python)을 선택하여 모듈을 설계하였으며 컴파일러 모듈의 언어 수정을 거치면 다른 언어의 처리도 가능하다.

코드 검증 모듈(Code check module)은 학습자가 작성한 코드가 잘 작성이 되었는지 체크해주는 기능을

하며, 미리 입력해둔 연습문제에 대한 테스트 데이터셋으로 채점이 된다. 코드의 정 오답의 여부는 시각적 도구가 팝업되어 표시된다.

코드 리뷰 모듈(Code review module)은 대화형 챗봇 형태로 설계되었으며, 대화 맥락이 반영되어 학습자의 코드의 변화 양상이 고려된 피드백 및 질의응답을 제공한다. 이 모듈은 대표적인 LLM인 GPT 모델을 기반으로 프롬프트 엔지니어링을 통해 구축되었으며 모듈의 역할과 제한사항이 프롬프트로 입력되었다. 먼저, 이 모듈이 예시 문제에 대한 학습자의 코드에 대한 리뷰 및 그에 대한 피드백을 제공할 수 있도록 역할을 부여하는 프롬프트가 입력되었다. 이때 초보 학습자를 대상으로 하는 한다는 점에서 제시하는 어휘의 난이도를 초급수준으로 설정하였다. 또한, 예시 답안 코드의 생성을 제한하는 프롬프트가 입력되어 학습자의 학습을 저해하는 경우를 방지하고자 하였다.

데이터베이스 구축에는 마이애스큐엘(MYSQL)이 활용되어 학습자의 로그데이터를 수집하고 보관할 수 있게 구성되었다. 수집되는 데이터는 학습자가 작성한 코드, 피드백 내용, 질문과 응답 등이 포함된다.

프론트 엔드에서 서버는 구글 클라우드 플랫폼(Google cloud platform)을 통해 구축되었으며, 웹 환경 설계에는 리액트(React)를 이용한 반응형 웹 환경을 기반으로, 앤트 디자인(Ant design) 기반의 오픈 소스를 통해 구축되었다.

### 3.4 사용자 평가 및 개선

본 연구에서 시스템을 검증하기 확인하기 위하여 총 3차례에 걸쳐 사용자 평가를 진행하였다. 평가자는 본 시스템이 고등학교와 대학교에서 활용될 가능성이 높다는 점에서 고등학교 교사와 대학생을 평가 대상으로 선정하였으며 컴퓨터 교육 전문가도 평가 대상으로 포함하였다. 이에 1차 사용자 평가에는 대학생 4인, 2차 평가에서는 컴퓨터 교육 전공 박사 3인 및 박사과정 2인, 3차 평가에서는 고등학교 정보 교사 3인으로 총 12인이 평가에 참여하였다.

평가는 참여자들이 본 학습 환경을 충분히 사용해 본 뒤 인터뷰를 통해 진행되었다. 평가 항목은 이지은(2021)이 제안한 온라인 학습 환경 평가지표가 본 연구에서 제안하는 시스템의 평가에 적용하기에 적합한 것으로 판단하여 선정하였다[24]. 이에 따라 ‘시스템 품질’, ‘서비스 품질’, ‘정보 품질’ 세 가지 영

**Table 1.** User evaluation and improvement directions

Construct	Index	Comments	Improvement directions
System quality	Easy installation and accessibility	-	-
	Convenient operation	The operation of each buttons are not clear	Modified buttons working only under certain conditions
	Quick and accurate response to learner behavior	Students may want to compile without feedback	Separating the modules
		Resource issue when multiple users access simultaneously	
Service quality	Intuitive and attractive UI	Student-centered UI design is needed	Add avatar to the interactive interface
			Show loading screen
			Presenting students' names in the chat history
	Sufficient study guide	Unifying the language is necessary	Default in Korean, with an option to choose English
		The exercise needs to be effectively expressed	Relocating the exercise instructions to the upper UI and adding a pop-up for additional details
		Presenting the sample code	Presenting the structure of sample code
	Providing feedback according to learner achievement	-	-
Information quality	Accuracy of information	Information from feedback sometimes miss some errors, especially syntax errors	Result from the code check or compiler module added to the prompt
	Providing customized learning information	-	-
	Readability	Feedback provided is usually too long	Modified the prompt to restrict the length of the feedback

역에 대하여 평가를 진행하였다. 이중 정보 품질은 시스템이 생성하는 피드백의 품질을 평가하는 지표로 사용되었다. 사용자 평가 결과에 따라 본 연구의 시스템 재설계 방향을 정리하였으며 정리된 내용은 Table 1과 같다.

시스템 품질과 관련하여서는 동시접속 문제와 컴파일 할 때마다 피드백과 코드 체크가 자동으로 되어서 오래 기다려야 하는 문제가 있었다. 이를 개선하기 위하여 학습자가 각 모듈을 개별적으로 사용할 수 있도록 각각의 모듈을 분리하였다. 또한, 각 모듈을 작동시키는 버튼들이 특정 조건에서만 동작할 수 있도록 재구성하였다. 코드 검증 모듈을 활성화하는 버튼과 코드 리뷰 모듈의 피드백 기능을 활성화하는 버튼은 학습자가 코드를 작성했을 때만 활성화되게 수정되었

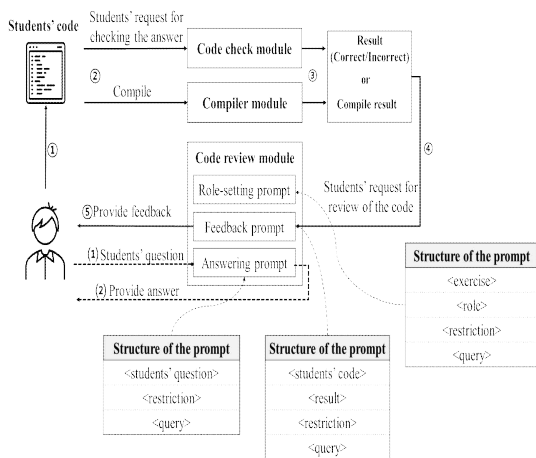
으며, 코드가 실행되고 있는 동안에는 다시 비활성화된다. 사용자의 입력을 받는 버튼은 input 함수가 사용되는 경우에만 활성화되도록 수정되었다.

서비스 품질과 관련하여서는 학습자 친화적인 UI로의 개선이 필요하다는 의견이 있었다. 이를 개선하기 위하여 코드 리뷰 모듈로 구현된 대화형 챗봇에 학습자의 친밀감을 유도하기 위한 이미지를 삽입하였고, 학습자의 채팅 부분에는 학습자의 이름이 표시되도록 수정하였다. 또한, 챗봇이 응답을 생성할 때까지 표시해주는 로딩 화면을 추가하였다. 국문과 영어가 혼재되어 있던 UI의 언어를 국문으로 통일하였고, 대화형 챗봇 부분은 영어로 변경할 수 있는 옵션을 추가하였다. 그리고 학습 가이드가 더 명확히 될 수 있도록 문제가 잘 보일 수 있는 부분으로 위치를 옮겼으며, 문

제에 대한 예시 입력 예시 출력과 예시 코드 구조를 보여주는 팝업창을 추가하였다.

정보 품질과 관련하여서는 코드 리뷰 모듈이 제시하는 피드백과 응답에 대한 평가가 이루어졌다. 이와 관련하여 문법 오류에 대한 피드백이 정확하지 않은 것을 확인하였고, 피드백의 길이가 너무 길어 가독성이 떨어지는 문제점을 파악하였다. 이를 개선하기 위하여 모듈의 프롬프트를 수정하였다.

Figure 3은 위와 같은 방법으로 개선된 시스템의 구성과 프롭트 프로토콜을 묘사하고 있다.



**Figure 3.** System flow and prompt protocol

먼저, 이 모듈의 역할을 부여하는 프롬프트가 입력된다. 여기에서 연습문제와 예시 답안이 입력되고, 그 다음으로 이 모듈의 역할과 제한사항이 입력된다. 모듈의 역할로 코드 리뷰와 질의응답이 설정되었으며, 학습자가 이 기능들을 요청할 때까지 대기상태로 기다린다.

각각의 기능은 별도의 프롬프트가 입력되어 응답이 생성된다. 학습자가 코드 작성을 하고 컴파일 및 코드 검증 모듈을 실행하다가 코드 리뷰가 필요하여 이를 요청하는 버튼을 활성화하는 경우, 코드 리뷰를 생성하기 위한 프롬프트가 자동으로 입력된다. 이 프롬프트에는 먼저 학습자의 코드가 입력되며, 컴파일러 모듈을 통한 컴파일 결과 혹은 코드 검증 모듈을 통해 확인된 정 오답 여부가 입력된다. 그리고 피드백 생성 시 제한사항이 입력되며 마지막으로 이를 요청하는 프롬프트가 입력된다. 제한사항으로 학습자가 문제를 스스로 풀어볼 수 있도록 정답 코드 생성의 제한과 가독성을 위한 3~4줄의 적절한 분량의 피드백 등이 입

력되었다.

또한, 학습자가 코드를 작성하는 과정 중에 질문을 입력하면 이에 대한 응답을 생성하기 위한 프롬프트가 자동으로 입력된다. 이 프롬프트에는 학습자의 질문이 먼저 입력이 되며, 학습자의 수준을 고려한 응답이 가능하도록 응답에 대한 난이도가 제한사항으로 입력되고 이에 대한 응답을 요청하는 프롬프트가 입력이 된다.

이 모듈은 각각의 응답을 마친 뒤 다시 대기상태로 돌아가는데, 이때 학습자와의 이전 대화 맥락이 계속 반영되어 학습자의 코드의 변화 양상과 질문 양상이 고려된 응답을 제공하게 된다.

수정된 시스템에 대하여 사용자들은 시스템이 학교 현장 안팎에서 프로그래밍 교육에 적합하다고 평가하였다.

“프로그래밍 수업 시 개별 피드백을 제공하기 어렵는데, 개발된 시스템은 학생의 코드를 분석하고 피드백을 제공해줌으로써 프로그래밍 수업에 매우 유용한 도구로 사용될 수 있을 것으로 기대되며, 자습할 때도 도움이 될 것 같아 보인다.” (교사 1)

“별도의 설치가 필요하지 않고 접속이 간편하여 학습자들에게 쉽게 배포할 수 있을 것으로 보이고 쉽게 사용 가능해 보인다.” (교사 2)







또한, 본 시스템에서 제공하는 피드백과 응답이 프로그래밍 학습에 유용하다고 평가하였으며, LLM을 교육에 접목하는 데 가이드라인이 될 수 있다고 평가하였다.

“선생님이나 강사님에게 질문하기 어려울 때 이 시스템을 효과적으로 사용할 수 있을 것 같다.” (대학생 1)

“프로그래밍 에러를 찾고 고치는 것은 매우 어려운 일인데, 피드백을 제공해주어서 고치가 훨씬 수월하다.” (대학생 2)

“GPT와 같은 LLM을 학교 현장에 사용할 수 있는 좋은 예시인 것 같다.” (컴퓨터 교육 박사 1)

Table 2. Validation and examples of feedback and responses

Construct	Mean (Std)				Code/ Students' question	Feedback/ Response
	Appropriateness	Specificity	Clarity	Usefulness		
Syntax error	4.27 (0.64)	4.33 (0.71)	4.10 (0.80)	4.23 (0.68)	<pre>N = int(input()) add_result = 0 count = 0 while N &gt; add_result: count += 1 add_result += count print(count)</pre>	 코드의 구조는 좋아요. 하지만 count += 1과 add_result += count가 들여쓰기를 안 해서 IndentationError가 발생했어요. 들여쓰기를 해서 코드를 수정해봐요.
Runtime error	4.30 (0.75)	4.27 (0.74)	4.17 (0.75)	4.33 (0.71)	<pre>N = input() add_result = 0 count = 1 while N &gt; add_result: count += 1 add_result += count print(count)</pre>	 좋은 시도였지만, N을 input()으로 받았기 때문에 N은 str 형식입니다. while 조건문에서는 N과 add_result를 비교하기 때문에 TypeError가 발생했습니다. 이를 해결하기 위해서는 N을 int로 형변환해주어야 합니다.
Logic error	4.10 (0.66)	4.23 (0.86)	4.13 (0.73)	4.37 (0.72)	<pre>N = int(input()) add_result = 0 count = 1 while N &gt; add_result: count += 1 add_result += count print(count)</pre>	 잘 노력했지만 코드에 한 가지 문제가 있네요. count를 초기값으로 1로 설정해줬는데 이렇게 되면 add_result가 1부터 시작하는 것이 아니라 2부터 시작하게 됩니다. 따라서 count를 0으로 초기화해주고, while문 안에서 더해주는 것이 좋겠어요.
Variable name	4.30 (0.60)	4.23 (0.77)	4.10 (0.61)	4.30 (0.65)	<pre>input = int(input()) add_result = 0 count = 0 while input &gt; add_result: count += 1 add_result += count print(count)</pre>	 코드가 잘 작성되었어요! 하지만 입력받은 정수를 저장하는 변수명이 'input'으로 지정되어 있습니다. 이는 파이썬에서 이미 함수로서 사용되는 이름이기 때문에, 코드에 오류가 발생할 가능성이 있습니다. 변수명을 다른 것으로 지정하면 더 좋을 것 같아요. 다른 부분은 문제를 잘 해결하고 있어요.
Unnecessary part	4.13 (0.68)	4.30 (0.75)	4.17 (0.70)	4.17 (0.79)	<pre>N = int(input()) add_result = 0 count = 0 while N &gt; add_result: count += 1 j=1 add_result += count print(count)</pre>	 코드를 보니 'j=1'이라는 코드는 삭제해도 되겠네요. 이 부분을 삭제해주시고, 입력받은 정수와 관련된 부분은 유지하면서 문제를 해결하시면 됩니다. 그리고 변수명도 적절하게 지어져 있어서 가독성도 좋습니다. 칭찬해요!
Response to the question	4.23 (0.73)	4.13 (0.78)	4.13 (0.78)	4.33 (0.66)	<p>“입력한 수가 홀수인지 짝수인지 어떻게 확인할 수 있을까?”</p>	 입력한 수가 홀수인지 짝수인지 알고 싶다면, 해당 수를 2로 나눈 나머지를 구하면 됩니다. 만약 나머지가 0이면 짝수이고, 1이면 홀수입니다. 나머지를 구하는 방법은 % 기호를 사용하면 됩니다. 예를 들어, n % 2 == 0 이면 짝수이고, n % 2 == 1 이면 홀수입니다.

### 3.5 피드백 및 응답의 검증과 예시

위와 같이 시스템을 개선한 뒤 피드백 및 응답의 적절성 평가를 진행하였다. 평가한 항목은 코드의 오류와 관련하여서는 ‘문법 오류’, ‘런타임 오류,’ ‘논리 오류’가 포함되었다. 이 오류들에 대한 피드백을 검증하기 위해 각각의 예제가 의도적으로 포함된 코드를 실행시키고 피드백 생성을 하였다.

학습자의 프로그래밍 습관과 관련한 피드백은 ‘변수명’, ‘불필요한 코드’가 포함되었으며, ‘학습자 질문에 대한 응답’도 평가되었다.

각각의 피드백 및 응답은 ‘적절성’, ‘구체성’,

‘명확성’, ‘유용성’ 네 가지 평가 준거에 대하여 평가되었다. ‘적절성’은 학습자가 개선해야 할 부분이나 학습자의 질문 내용과 관련된 적절한 내용의 응답을 하는지 확인하는 준거이며, ‘구체성’은 피드백과 응답이 구체적으로 제시되어 학습자의 학습을 효과적으로 지원하는지 확인하는 준거이다. ‘명확성’은 시스템에서 제공하는 피드백 및 응답이 명확하게 제시되어 학습자가 쉽게 읽고 이해할 수 있는지 확인하는 구인이며, ‘유용성’은 학습자의 학습에 도움이 되는지 확인하는 구인이다.

각각의 구인은 리커트 5점 척도로 교사 3인에 의해 평가되었으며, 베이스가 된 GPT 모델의 생성적 성격

에 따라 항목별로 10회 평가하여 점수의 평균을 구하였다. Table 2에는 이와 같은 방법으로 진행된 평가 결과와 시스템의 응답 예시가 제시되어 있다.

### 3.6. UI 및 실행 예시

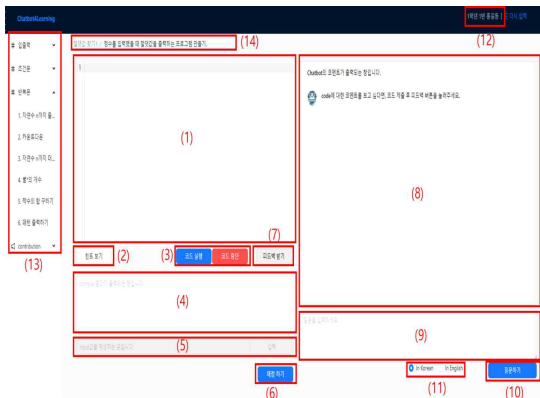


Figure 4. User Interface

Figure 4는 개발된 시스템의 UI를 나타내고 있으며, 각 부분의 역할과 기능은 아래와 같다.

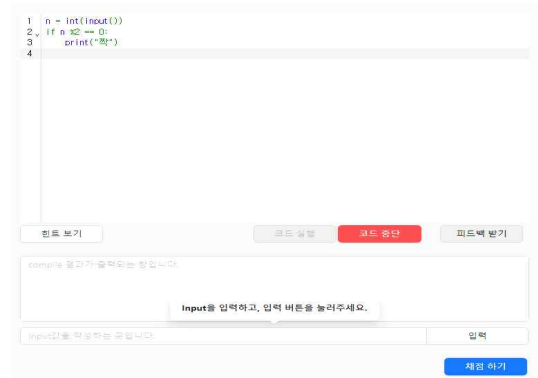
- (1) 학습자가 코드를 작성할 수 있는 공간이다.
- (2) 주어진 연습문제에 세부 정보를 확인할 수 있는 공간이다. 예시 입력, 예시 출력, 예시 코드의 구조 등이 표시된다.
- (3) 학습자가 작성한 코드를 실행시키고 중단할 수 있는 버튼이다.
- (4) 학습자가 작성한 코드의 컴파일 결과를 확인할 수 있는 공간이다. 코드 실행 결과 에러가 발생한다면 발생한 에러가 출력된다.
- (5) 학습자가 input() 함수를 사용하는 경우 입력값을 작성하고 제출할 수 있는 기능을 한다. input() 함수가 실행될 때만 해당 기능이 활성화되며 일정 시간이 지나면 자동으로 종료되고 다시 비활성화된다.
- (6) 코드 검증 모듈을 활성화하는 버튼으로 학습자가 작성한 코드를 채점해주는 기능을 한다.
- (7) 코드 리뷰 모듈의 피드백 제공 기능을 활성화하는 버튼이다. (3)의 ‘코드 실행’ 버튼을 클릭하여 코드 실행이 완료되면 활성화되며, 코드가 실행되고 있는 동안에는 다시 비활성화된다.
- (8) 코드 리뷰 모듈의 피드백, 학습자의 질문과 그에 대한 응답이 표시되는 공간이다. 그동안의 누적된 대화 내용이 모두 제시된다. 학습자와 인공지능의 채팅 내용은 구분되어 표기된다.

(9) 학습자가 질문을 입력하는 공간이다.

(10) 코드 리뷰 모듈의 질의응답 기능을 활성화하는 기능을 하는 버튼이다.

(11) 채팅 및 피드백의 언어를 선택하는 기능을 하는 버튼이다.

한편, Figure 5은 문제 풀이 및 채점 예시를 나타내고 있으며, Figure 6은 대화형 챗봇과의 응답 예시를 나타내고 있다.



(a)Example of inserting an input value



(b)Example of result when a syntax error occurs.



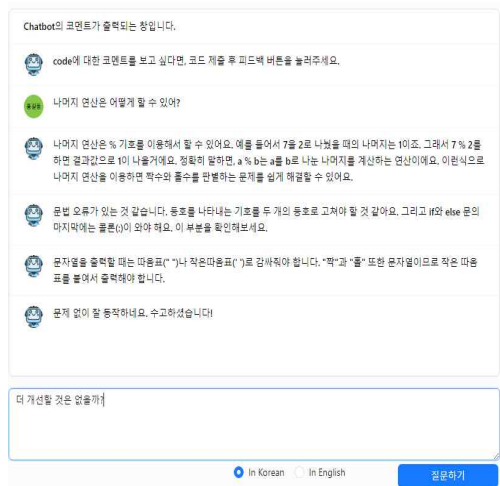
(c)Example of result when the student correctly solves the exercise



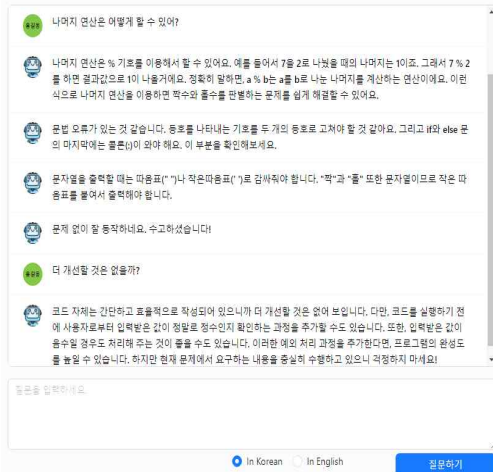
(d)Example of result when the student incorrectly solves the exercise

Figure 5. Example of code writing and evaluation





(a) Example of student's question input



(b) Example of review for the student's code and response to the student's question.

Figure 6. Examples of responses

## 4. 결론

본 연구에서는 프로그래밍 학습을 위한 LLM 기반 코드 리뷰 시스템을 개발하였다. 본 연구는 선행 연구 분석을 통해 설계 원리를 도출하고 이를 기반으로 다양한 개발 환경을 사용하여 웹 시스템을 구축하였고 사용자 평가를 거쳐 시스템을 개선하였다. 본 시스템은 대표적인 LLM인 GPT 모델을 기반으로 프롬프트 엔지니어링을 통해 구축되었으며, 제안된 프롬프트

프로토콜은 LLM을 프로그래밍 교육에 활용할 때의 가이드라인이 될 수 있을 것으로 보인다.

개발된 시스템은 연습문제 풀이에 있어 학습자의 다양한 접근과 변화되는 코드의 양상 및 맥락을 고려한 피드백을 제공하며 학습자는 관련한 질문을 하고 이에 대해 답변을 받으며 학습할 수 있다. 특히, 문법 오류뿐만 아니라 논리 오류에 대한 피드백과 코딩 습관과 관련한 피드백도 제공한다. 또한, 본 시스템은 컴파일 기능도 제공하고 있어서 학습자가 별도의 프로그래밍 환경이 필요로 하지 않는다. ChatGPT의 경우 자체적으로 컴파일이 가능하지 않으며 별도의 프롬프트를 입력하지 않으면 주어진 연습문제의 정답을 바로 생성해버리는 문제가 있다는 점에서, 본 시스템은 ChatGPT를 직접적으로 활용하는 것보다 프로그래밍 학습에 더욱 유의미할 것으로 보인다.

본 시스템은 학교 현장에서 프로그래밍 학습에 효과적인 수업 도구로 사용 가능하며, 맞춤형 프로그래밍 학습에 유용한 도구로 활용될 수 있을 것으로 보인다. 후속 연구로는 본 시스템을 학생들에게 적용했을 때 학생들의 프로그래밍 학습에 효과적인 영향을 주는 지 검증하는 연구가 진행될 필요가 있다. 개발된 시스템은 <http://chatbot4learning.com/>에서 확인할 수 있다.

## 참고문헌

- [ 1 ] Howard, S. K., Tondeur, J., Ma, J., & Yang, J. (2021). What to teach? Strategies for developing digital competency in preservice teacher training. *Computers & Education*, 165, 104149. <https://doi.org/10.1016/j.compedu.2021.104149>
- [ 2 ] Dondi, M., Klier, J., Panier, F., & Schubert, J. (2021). Defining the skills citizens will need in the future world of work. *In Public & Social Sector Practice* (Issue June). <https://mck.co/3b1wULK>
- [ 3 ] Gallardo-Echenique, E. E., de Oliveira, J. M., Marqués-Molias, L., Esteve-Mon, F., Wang, Y., & Baker, R. (2015). Digital competence in the knowledge society. *MERLOT Journal of Online Learning and Teaching*, 11(1).
- [ 4 ] Oberländer, M., Beinicke, A., & Bipp, T. (2020). Digital competencies: A review of the literature and applications in the workplace. *Computers & Education*, 146, 103752. <https://doi.org/10.1016/j.compedu.2019.103752>
- [ 5 ] Kim, H. S., Jun, S., Choi, S., & Kim, S. (2020). Development and application of education program on understanding artificial intelligence and social

- impact. *The Journal of Korean association of computer education*, 23(2), 21-29. <https://doi.org/10.32431/kace.2020.23.2.003>
- [ 6 ] Jang, Y., Choi, S., Cho, H., & Kim, H. (2022). Development and Application of Modular Artificial Intelligence Ethics Education Program for Elementary and Middle School Students. *The Journal of Korean association of computer education*, 23(5), 1-14. <https://doi.org/10.32431/kace.2022.25.5.001>
- [ 7 ] Jang, Y., Choi, S., Kim, S., & Kim, H. (2023). The SNS-based E-mentoring and Development of Computational Thinking for Undergraduate Students in an Online Course. *Educational Technology & Society*, 26(2), 147-164. [https://doi.org/10.30191/ETS.202304\\_26\(2\).0011](https://doi.org/10.30191/ETS.202304_26(2).0011)
- [ 8 ] Radosevic, D., Lovrencic, A., Orehovacki, T.: New Approaches and Tools in Teaching Programming. In: *Central European Conference on Information and Intelligent Systems* (2009)
- [ 9 ] Keuning, H., Jeuring, J., & Heeren, B. (2018). A systematic literature review of automated feedback generation for programming exercises. *ACM Transactions on Computing Education (TOCE)*, 19(1), 1-43. <https://doi.org/10.1145/3231711>
- [ 10 ] Gerdes, A., Heeren, B., Jeuring, J., & Van Binsbergen, L. T. (2017). Ask-Elle: an adaptable programming tutor for Haskell giving automated feedback. *International Journal of Artificial Intelligence in Education*, 27, 65-100. <https://doi.org/10.1007/s40593-015-0080-x>
- [ 11 ] Crow, T., Luxton-Reilly, A., & Wuensche, B. (2018, January). Intelligent tutoring systems for programming education: a systematic review. In *Proceedings of the 20th Australasian Computing Education Conference* (pp. 53-62). <https://doi.org/10.1145/3160489.3160492>
- [ 12 ] Kasneci, E., Seßler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., ... & Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103, 102274. <https://doi.org/10.1016/j.lindif.2023.102274>
- [ 13 ] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., ... & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35, 24824-24837.
- [ 14 ] Badampudi, D., Britto, R., & Unterkalmsteiner, M. (2019). Modern code reviews - Preliminary results of a systematic mapping study. *ACM International Conference Proceeding Series*, 340-345. DOI : <https://doi.org/10.1145/3319008.3319354>
- [ 15 ] Indriasari, T. D., Luxton-Reilly, A., & Denny, P. (2020). A Review of Peer Code Review in Higher Education. *ACM Transactions on Computing Education*, 20(3), 22. DOI : <https://doi.org/10.1145/3403935>
- [ 16 ] Sebastian Gross, Bassam Mokbel, Barbara Hammer, and Niels Pinkwart. 2015. Learning feedback in intelligent tutoring systems. *Künstliche Intelligenz* 29, 4 (2015), 413-418. <https://doi.org/10.1007/s13218-015-0367-y>
- [ 17 ] Suarez, M., & Sison, R. (2008). Automatic construction of a bug library for object-oriented novice java programmer errors. In *Intelligent Tutoring Systems: 9th International Conference, ITS 2008, Montreal, Canada, June 23-27, 2008 Proceedings* 9 (pp. 184-193). Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-540-69132-7\\_23](https://doi.org/10.1007/978-3-540-69132-7_23)
- [ 18 ] Choi, S., Jang, Y., & Kim, H. (2023). Influence of pedagogical beliefs and perceived trust on teachers' acceptance of educational artificial intelligence tools. *International Journal of Human-Computer Interaction*, 39(4), 910-922. <https://doi.org/10.1080/10447318.2022.2049145>
- [ 19 ] Jang, Y., Choi, S., Jung, H., & Kim, H. (2022). Practical early prediction of students' performance using machine learning and explainable AI. *Education and Information Technologies*, 1-35. <https://doi.org/10.1007/s10639-022-11120-6>
- [ 20 ] Choi, S., Jang, Y., & Kim, H. (2022). A Deep Learning Approach to Imputation of Dynamic Pupil Size Data and Prediction of ADHD. *International Journal on Artificial Intelligence Tools*. <https://doi.org/10.1142/S0218213023500203>
- [ 21 ] Jeon, J., & Lee, S. (2023). Large language models in education: A focus on the complementary relationship between human teachers and ChatGPT. *Education and Information Technologies*, 1-20. <https://doi.org/10.1007/s10639-023-11834-1>
- [ 22 ] Chiu, T. K., & Churchill, D. (2015). Exploring the characteristics of an optimal design of digital materials for concept learning in mathematics: Multimedia learning and variation theory. *Computers & Education*, 82, 280-291. <https://doi.org/10.1016/j.compedu.2014.12.001>
- [ 23 ] Choi, S., Jang, Y., & Kim, H. (2023). Exploring factors influencing students' intention to use intelligent personal assistants for learning. *Interactive Learning Environments*, 1-14. <https://doi.org/10.1016/j.compedu.2014.12.001>

i.org/10.1080/10494820.2023.2194927

[ 24 ] Lee, J. (2021). Development of LMS Evaluation Index for Non-Face-to-Face Information Security Education. *Journal of The Korea Institute of Information Security and Cryptology*, 31(5), 1055-1062. <https://doi.org/10.13089/JKIISC.2021.31.5.1055>

### 최 승 윤



2016년 경인교육대학교  
초등컴퓨터교육과 (교육학학사)  
2019년 경인교육대학교  
초등컴퓨터교육과 (교육학석사)  
2022년 고려대학교 일반대학원  
컴퓨터학과 (박사 수료)

2016년~현재 초등학교 교사  
관심분야: CS/AI 교육, AIED, 교육 공학  
E-Mail: csyune213@korea.ac.kr

### 이 동 건



2022년 고려대학교 경제학과 (학사)

2022년 ~ 현재 고려대학교 일반대학원 컴퓨터학과 (석사과정)  
관심분야: 기계학습, 강화학습  
E-Mail: pocet25@korea.ac.kr

### 김 준 구



2020년 고려대학교 컴퓨터학과 (학사)

2022년 ~ 현재 고려대학교 일반대학원 컴퓨터학과 (석사과정)  
관심분야: 자연어처리  
E-Mail: antonio97k@korea.ac.kr

### 장 연 주



2013년 서울교육대학교  
초등컴퓨터교육과 (교육학학사)  
2019년 서울교육대학교  
초등컴퓨터교육과 (교육학석사)  
2023년 고려대학교 일반대학원  
컴퓨터학과 (공학 박사)

2023년~현재 고려대학교 정보창의교육연구소  
관심분야: CS/AI 교육, 학습 분석, AIED  
E-Mail: spring0425@korea.ac.kr

### 김 현 철



1988년 고려대학교 전산과학과(학사)  
1990년 Univ of Missouri-Rolla  
(전산학석사)  
1998년 Univ of Florida (전산정보학 박사)

1999년~현재 고려대학교 컴퓨터학과 교수  
관심분야: SW/AI 교육, 기계학습  
E-Mail: harrykim@korea.ac.kr