Step With Me

스텝 윗 미

2024학년도 2학기 소프트웨어공학개론

5조: 강은비 박지연 신윤성 이일규 이채헌 정정환

TABLE OF CONTENTS

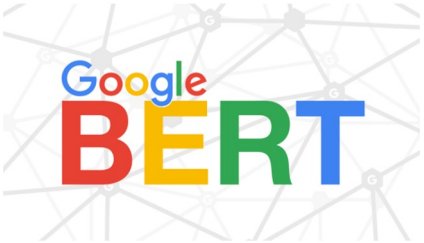
01

Background

What are LLMs?

- 수많은 데이터를 학습하여 인간 언어를 이해하고 처리하는 인공지능 모델
- 방대한 양의 텍스트를 통해 언어 패턴과 의미 학습, 이를 바탕으로 다양한 작업 수행

  ex) GPT, BERT 등

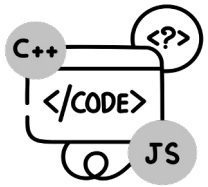


Why LLMs?

개인화된
학습 경험 제공

24/7
튜터링

다양한 학습
콘텐츠 생성

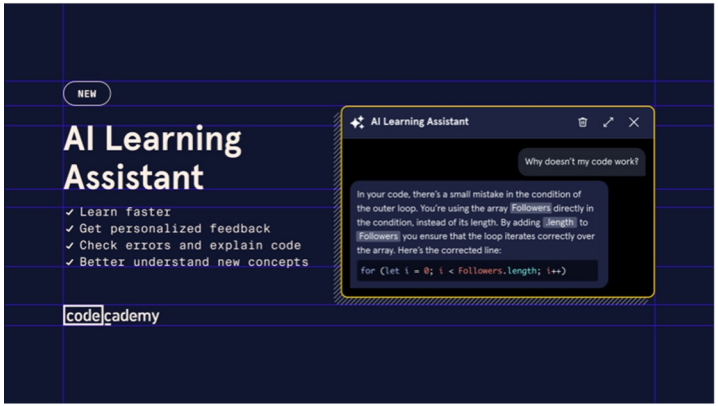


다양한
프로그래밍 언어
지원

생산 환경에서의
중요성

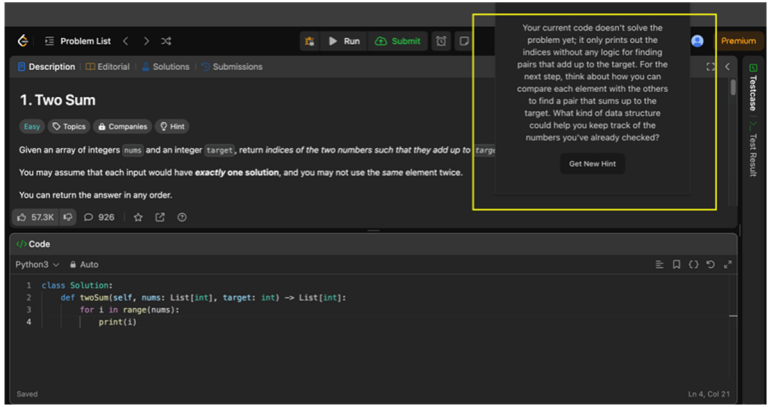
Application Examples

Codecademy



- 프로그래밍 학습 플랫폼
- AI: 실시간 코딩 피드백, 문제 해결 방법 제안

LeetCode

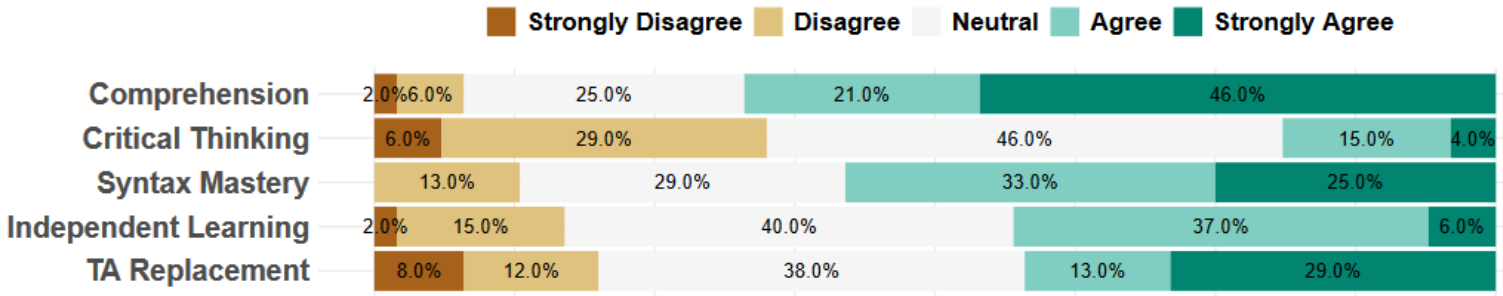


- 알고리즘 문제 풀이 사이트
- AI: 문제 해설, 코드 최적화 제안

Limits

LLM을 사용한 프로그래밍 교육의 한계

- 비판적 사고나 문제 해결 능력 향상 저해
- 스스로 문제의 논리 구조를 온전히 이해하지 못한 채 LLM에 의존 -> 학습의 질 저하



Participants' attitudes toward CodeTutor, in terms of comprehension, critical thinking, syntax mastery, independent learning, and TA replacement[1]

1: Lyu, W., Wang, Y., Chung, T. R., Sun, Y., & Zhang, Y. (2024). *Evaluating the Effectiveness of LLMs in Introductory Computer Science Education*. L@S '24. https://doi.org/10.1145/3657604.3662036

Step With Me

Step-by-step coding tutor

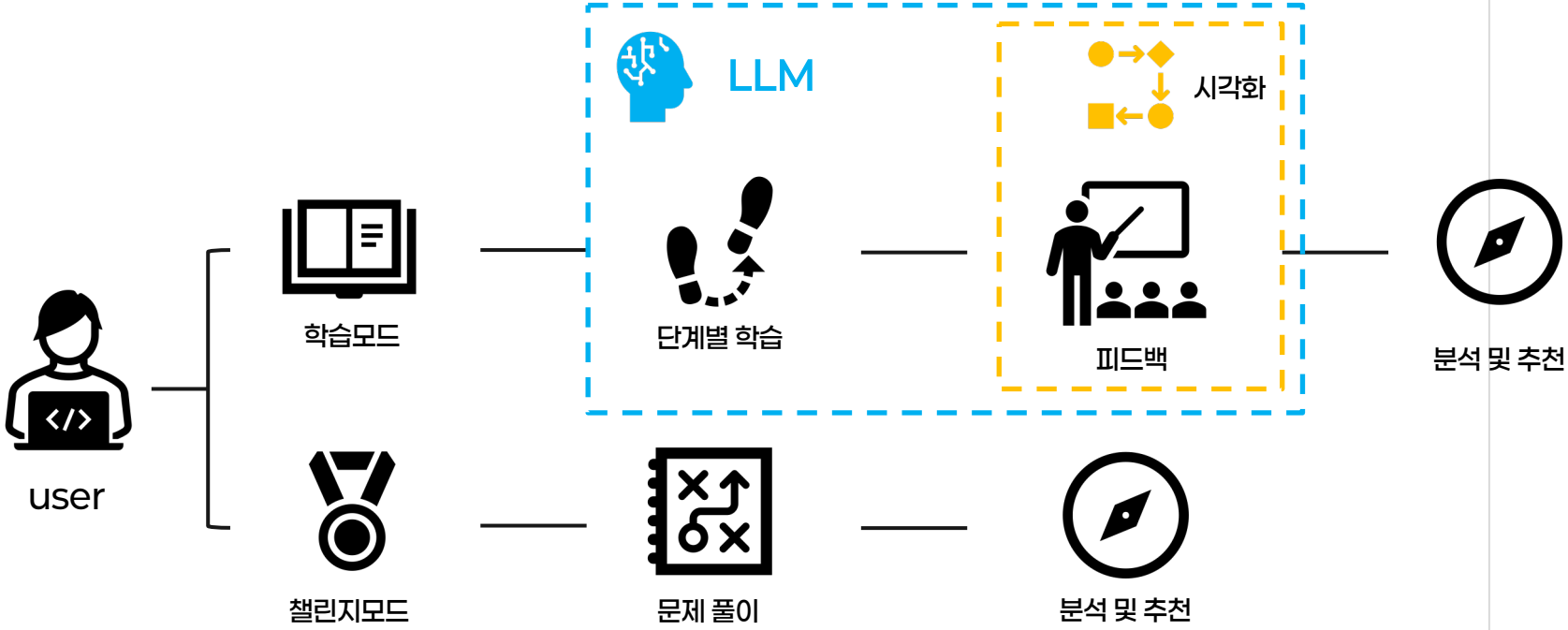
02

Goals & Features

Goals

- 학습자의 수준을 파악하여 맞춤형 교육 과정 제공
- 사용자의 수준을 고려한 단계별 학습과 알고리즘 시각화를 통한 학습 이해도 증진
- 개인화된 피드백 제공 및 즉각적 반영을 통한 학습자의 의욕 고취
- 분리된 학습 모드 속에서의 자기주도 학습 유도

Overview



Features

1️⃣ 밀착형 AI와의 단계별 학습

- Fine-tuning된 LLM을 이용한 문제 제공 및 수준별 단계화
- 핵심 논리 단위 별 가이드라인과 테스트케이스 생성
- 사용자 수준에 따라 각 단계의 분할 정도 상이
- 오류 로그, 단계별 안내 등에 자연어가 사용 : 학습자 친화적으로 진행

Feature Specs

알고리즘의 로직 단위 분할 및 세부 가이드라인 제공

- 핵심 논리 단위 별 분할: 알고리즘을 난이도에 따라 단계별로 분해
- 각 단계별 자연어 가이드라인, 채점결과 기반 피드백 제공

코드 평가의 자동화

- 코드 정합성 검사: 정적 분석 도구(예: pylint, flake8) 사용
- 유닛 테스트 생성: 자동으로 유닛 테스트 코드를 생성, 다양한 입력에 대해 작동 검증.
  - ✓ 유닛 테스트에서 모든 테스트 케이스를 통과해야만 최종 코드가 학생에게 전달

Features

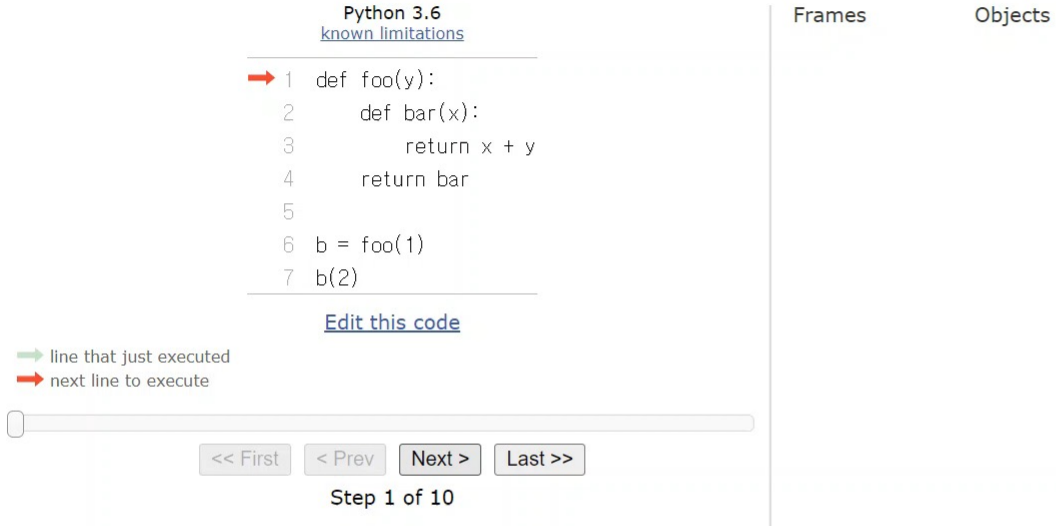
2️⃣ 오답에 대한 다각도 피드백 제공

- 문제 진행 중 오답, 에러 발생 시 작성한 알고리즘의 시각화를 통해 직관적인 이해 유도
- LLM이 사용자의 알고리즘과 에러 메세지를 분석하여 자연어 피드백 제공

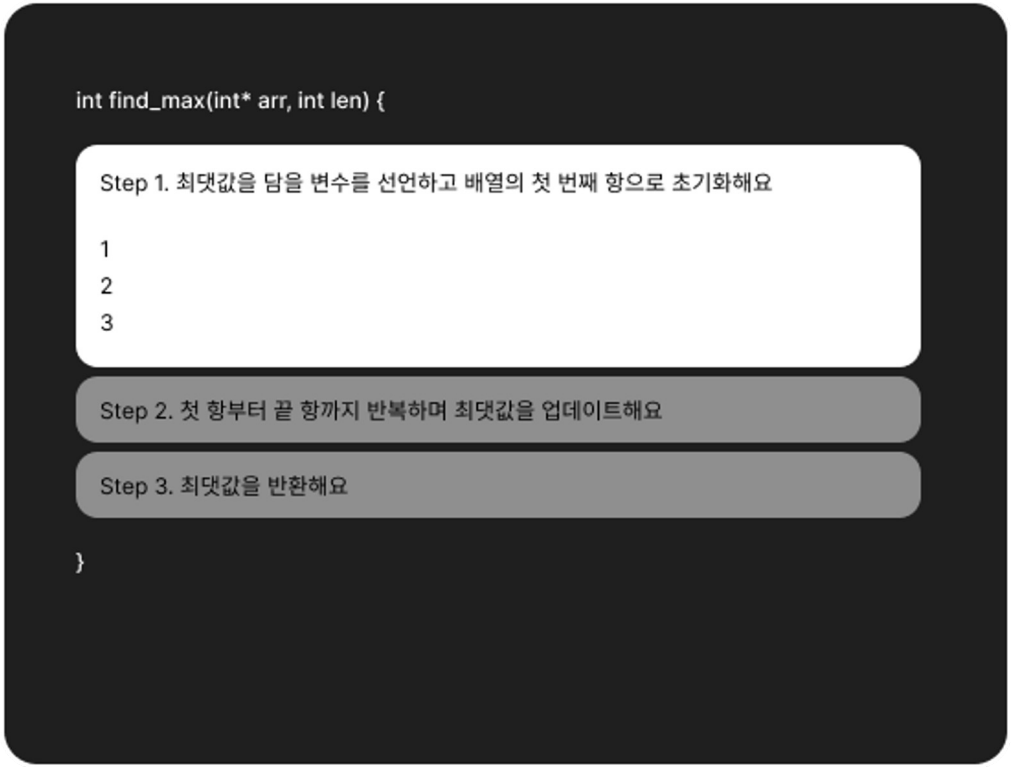
Feature Specs: Visualization

Python Tutor 오픈소스 활용

사용자의 입력 코드의 과정을 따라가는 시각화 자료 제공



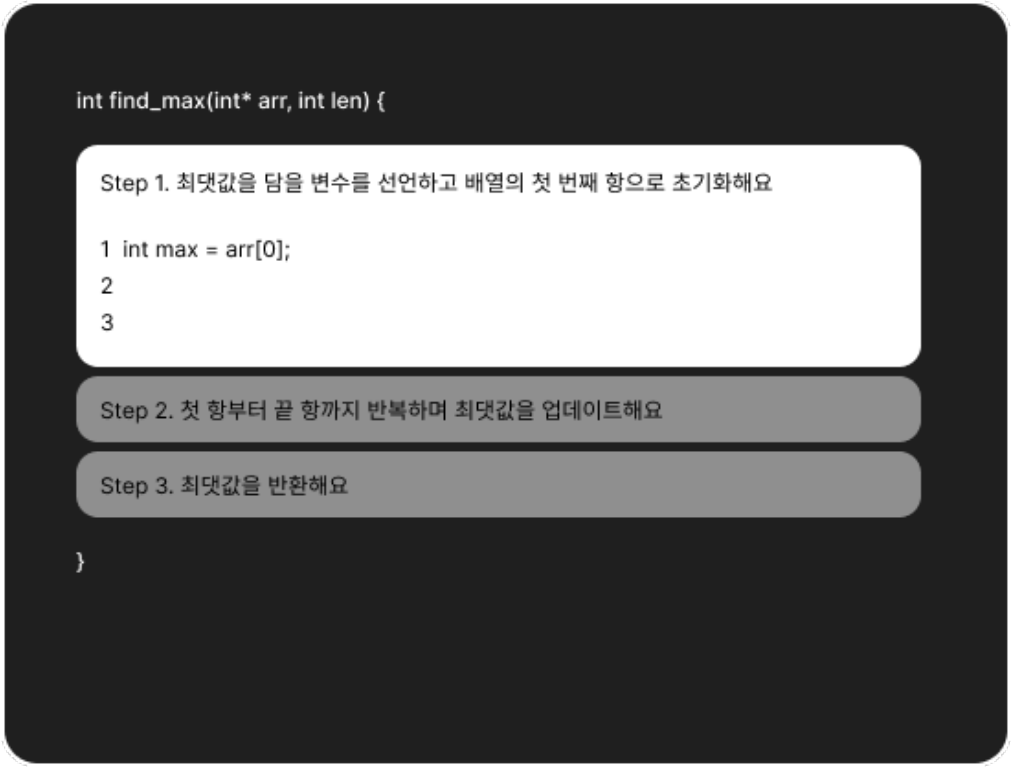
Wireframe



1개의 문제, 3개의 sub logic

핵심 논리 단위로 분해 제공
채점을 위한 test case는
사전 제작 등록

Wireframe



sub logic별 자연어 가이드라인

Wireframe

int find_max(int* arr, int len) {

Step 1. 최댓값을 담을 변수를 선언하고 배열의 첫 번째 항으로 초기화해요

```
1   int max = arr[0];
```

Step 2-1. 첫 항부터 끝 항까지 반복하는 for문을 작성해요

```
2   for(int i=0;i<len;i++){
3   }
4
```

Step 3. 최댓값을 반환해요

}

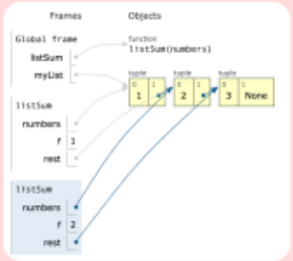
sub logic통과 시 다음으로 진행

Wireframe

```
int find_max(int* arr, int len) {

    Step 1. 최댓값을 담을 변수를 선언하고 배열의 첫 번째 항으로 초기화해요.

1   int max = arr[0];

2   for(int i=0;i<len;i++){
        Step 2-2. i번째 항과 max를 비교하고, 더 크다면 max값을 업데이트해요.
3       if(arr[i] > max) {
4           max == arr[i];
5       }

    Step 3. 최댓값을 반환해요

}
```

실행 코드에 대한
다각도 피드백 제공
(시각 자료, 자연어 피드백 등)

- max값이 업데이트되지 않았어요
- 비교 연산자(==)는 두 값이 동일한지 비교할 때 사용해요



Wireframe

```
int find_max(int* arr, int len) {
    Step 1. 최댓값을 담을 변수를 선언하고 배열의 첫 번째 항으로 초기화해요.
1   int max = arr[0];
    Step 2. 첫 항부터 끝 항까지 반복하며 최댓값을 업데이트해요
2   for(int i=0;i<len;i++) {
3     if(arr[i] > max) {
4     max = arr[i];
5   }
    Step 3. 최댓값을 반환해요
6   return max;
}
```

Features

3️⃣ 히스토리 기반 개인화 교육환경 제공

- 유형 별 오답률 등 사용자의 학습 데이터를 분석하여 추가 학습 과정 제시
- 학습 중 잦은 어려움을 겪은 부분에 대한 예제 추천
- 최종 코드 리뷰 제공, 타 사용자들과 비교 분석

Feature Specs

사용자 데이터 관리 및 최적화

- 작성한 코드, 오류 로그, 성능 분석 결과 등을 데이터베이스에 저장
- 피드백을 수집해 모델이 더 나은 코드 생성 능력을 갖출 수 있도록 학습
- Codebert Hugging Face의 라이브러리와 Database의 사용자 Dataset을 활용해 Fine-Tuning 진행 → 히스토리 기반 개인화 교육 환경 제공

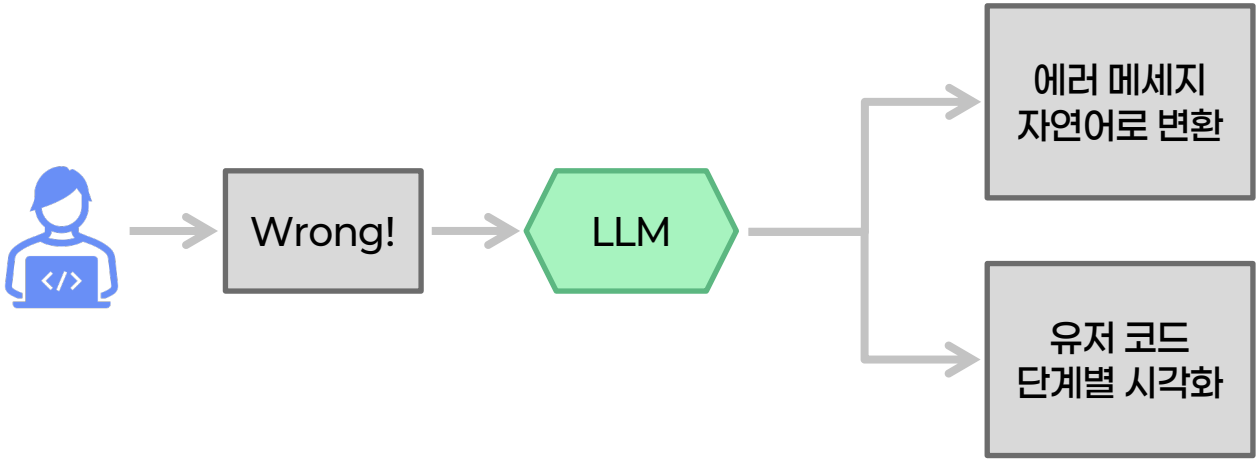
Features

4️⃣ 학습 모드와 챌린지 모드 제공

- 학습 모드: AI와의 밀착형 1:1 학습을 통하여 알고리즘 유형과 로직을 연습
- 챌린지 모드: AI의 도움 없이 실전 문제 해결

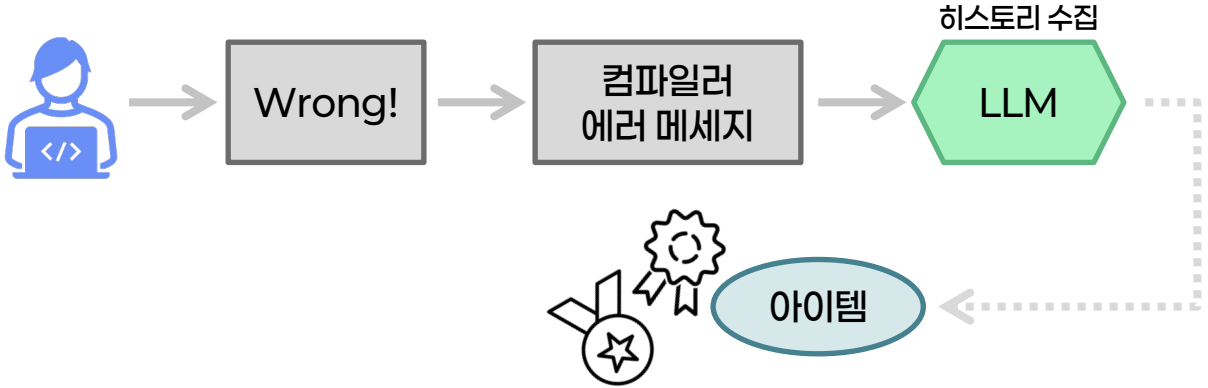
Feature Insight: Study Mode

- 문제와 로직 분할, 테스트케이스는 미리 내장된 상태로 제공
- 오답 또는 에러 시 유저 입력 코드를 단계별로 시각화 또는 자연어 피드백 제공
- 컴파일 에러: 컴파일러 에러 메세지를 LLM으로 자연어화
- 런타임 에러 : 시각화 및 LLM이 에러에 대한 자연어 피드백과 해결방안 제시



Feature Insight: Challenge Mode

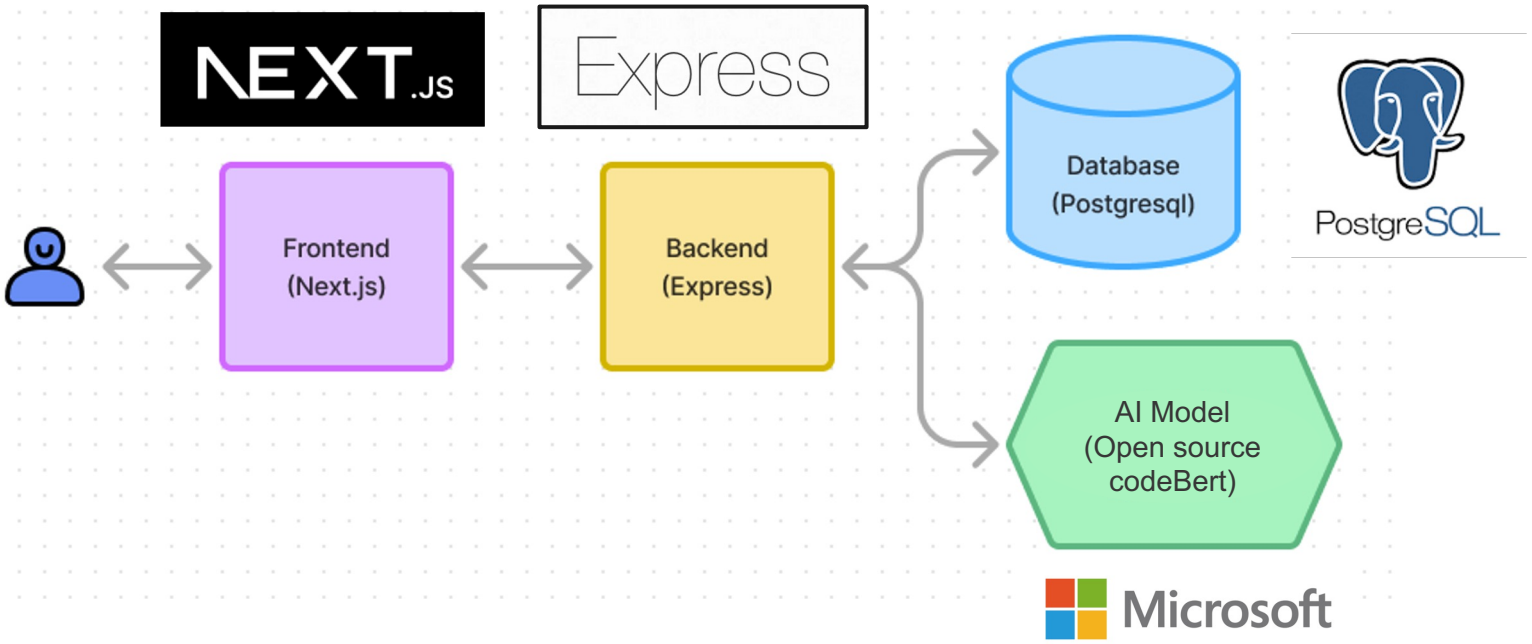
- 디버깅, 에러 메세지 등 최소한의 기능 제공
- 사용자 입력 정보 및 피드백 수집 및 분석은 동일하게 진행
- 강점 및 성취 분석을 통해 뱃지, 업적 등의 아이템 제공



03

Development

Overall architecture

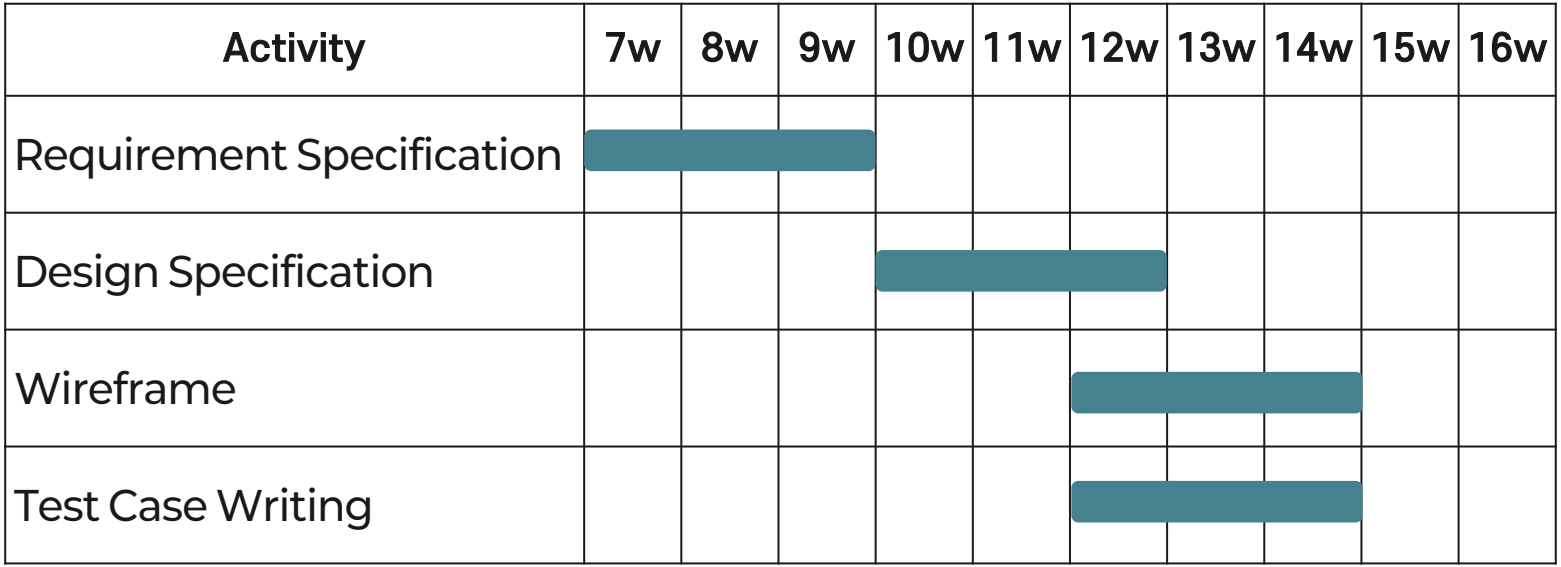


04

Roles & Plan

Role	Member(s)
총괄 기획	정정환
문서 작성 및 관리	신윤성
UI 디자인	박지연
DB 설계 및 개발	강은비
백엔드 서버 개발	이채헌
LLM 설계 및 관리	이일규

Overall Plan



THANK YOU!