

# Software Requirements Specification

## **Step With Me**

강은비, 박지연, 신윤성, 이일규, 이채현, 정정환

SWE3002-41

Team 5

# Contents

## **1 Introduction**

- 1.1 Purpose
- 1.2 Scope
- 1.3 Definitions, acronyms, and abbreviations
- 1.4 References
- 1.5 Overview

## **2 Overall Description**

- 2.1 Product perspective
- 2.2 Product functions
- 2.3 User classes and characteristics
- 2.4 Design and implementation constraints
- 2.5 Assumptions and dependencies

## **3 External Interface Requirements**

- 3.1 External interfaces
- 3.2 Function requirements
- 3.3 Performance requirements
- 3.4 Logical database requirements
- 3.5 Design constraints
- 3.6 Software system attributes
  - 3.6.1 Reliability
  - 3.6.2 Availability
  - 3.6.3 Security
  - 3.6.4 Maintainability
  - 3.6.5 Portability

## **4 Appendices**

# 1 Introduction

## 1.1 Purpose

이 문서는 초심자를 대상으로 프로그래밍 학습을 보조해주는 인공지능 프로그래밍 학습 프로그램 **Step With Me**의 요구사항을 명확히 하고, 소프트웨어의 개발자 또는 본 소프트웨어의 요구사항을 알고자 하는 자들을 위해 작성되었다.

## 1.2 Scope

소프트웨어 이름: **Step With Me**

소프트웨어의 역할:

1. 사용자의 수준과 원하는 목표를 파악하고 적당한 알고리즘 문제를 제시해준다.
2. 사용자의 선택에 따라 챌린지모드와 학습모드의 문제 풀이 환경을 제공한다.
3. 학습모드에서는 알고리즘을 세분화, 자연어 가이드라인, 시각적 피드백 등을 제공하며 문제 해결을 돕는다.
4. 챌린지모드에서는 인공지능의 도움 없이 독자적인 문제 풀이 환경을 제공한다.
5. 사용자의 오답 경향성, 알고리즘 문제 마일스톤 등의 히스토리를 기반으로 추천 문제를 업데이트한다.
6. 타 사용자들과의 비교 및 분석을 제공한다

소프트웨어를 통해 얻을 수 있는 이점, 목적 및 목표

- 프로그래밍에 있어 인공지능은 자동화 코드를 제공함으로써 많은 이점을 주지만, 사람들이 학습하는 과정에서 비판적인 사고나 문제 해결 능력 향상을 저해하는 부작용이 있다. 문제의 논리 구조와 세부적인 로직들을 이해하려 하지 않고 인공지능의 보조에 쉽게 의지하면 학습 동기와 성취도가 떨어질 수 있는데, 이 소프트웨어를 통해 이런 한계를 극복하고 자기주도 학습 환경 속에서 학습자의 의욕을 고취시키는 것을 목표로 한다. 또한 세분화되고 개인화된 피드백을 통해 학습 성취도를 증강한다.

## 1.3 Definitions, acronyms and abbreviations

Terms	Definitions
<b>LLM</b>	Large Language Model, 사용자 코드에 대한 실시간 피드백 및 문제 해결 가이드를 제공하기 위해 사용되는 대규모 언어 모델.
<b>Study mode</b>	AI 피드백을 통해 단계별로 문제를 풀며, 코드의 논리적 흐름을 학습하고 자신의 코드에 대한 개선점을 확인할 수 있는 학습 모드.
<b>Challenge mode</b>	AI의 도움 없이 문제 해결을 수행하며, 실전과 같은 환경에서 사용자 스스로 코드를 완성하도록 하는 모드.
<b>Feedback Panel</b>	학습 모드에서 사용자가 작성한 코드의 오류와 개선점을 시각적으로 제공하는 인터페이스.

Acronyms & Abbreviations	Explanation
<b>SRS</b>	Software Requirements Specification (소프트웨어 요구사항 명세서)
<b>API</b>	Application Programming Interface (애플리케이션 프로그래밍 인터페이스)
<b>DBMS</b>	Database Management System (데이터베이스 관리 시스템)
<b>HTTPS</b>	Hypertext Transfer Protocol Secure, 보안 웹 통신을 위한 프로토콜
<b>gRPC</b>	Google Remote Procedure Call, AI 모델과의 데이터 통신을 위한 효율적인 프로토콜
<b>CPU</b>	Central Processing Unit (중앙 처리 장치)
<b>RAM</b>	Random Access Memory (임의 접근 메모리)

## 1.4 References

IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications, In IEE- EXplore Digital Library

<http://ieeexplore.ieee.org/Xplore/guesthome.jsp>

pythonTutor

<https://pythontutor.com/>

## 1.5 Overview

본 SRS 문서는 3개의 섹션으로 구성된다.

첫 번째 섹션에서는 **Team 5**에서 제시하는 **Step With Me** 인공지능 프로그래밍 학습 소프트웨어 서비스와 그 목적에 대한 소개, 산출물의 범위, 문서 해석에 필요한 용어, 참조 문서, 그리고 문서의 구성에 대해서 다룬다.

두 번째 섹션에서는 소프트웨어의 전체적인 조망, 제품의 기능, 사용자의 특성, 제약사항, 가정 및 의존성, 그리고 요구사항 분배에 대하여 서술한다.

마지막 섹션에서는 소프트웨어의 외부 인터페이스 요구사항, 시스템 특징과 성능 요구사항을 기술한다.

# 2 Overall Descriptions

## 2.1. Product Perspective

"Step With Me"는 학습자가 자신의 수준에 맞는 맞춤형 코딩 학습을 할 수 있도록 지원하는 LLM 기반 튜터링 시스템이다. 기존 플랫폼의 일방향적 피드백 한계를 극복하고, 실시간 맞춤형 피드백을 제공하여 학습자의 이해도와 학습 성취감을 높이는 것을 목표로 한다. 이 시스템은 LLM을 활용해 수준에 맞는 단계별 학습 문제를 제공하고 실시간 피드백과 알고리즘 시각화를 통해 학습자의 이해도를 높인다. 또한, 자기주도적 학습을 장려해 학습자가 스스로 내용을 분석하고 개선할 수 있는 역량을 키울 수 있다.

### 2.1.1. System Interfaces

1. 데이터베이스 인터페이스: 학습자의 진행 상황, 피드백, 문제 풀이 결과 등을 저장하고 조회할 수 있도록 시스템의 데이터베이스와 상호작용한다. 데이터베이스에 저장된 사용자의 문제 풀이 히스토리를 기반으로 맞춤형 문제를 추천한다.
2. AI 모델 인터페이스: LLM을 활용하여 학습자의 수준에 맞는 문제를 생성하거나 피드백을 제공한다. 사용자가 문제를 제출하면 시스템은 LLM과 통신하여 맞춤형 피드백과 해설을 제공한다.

### 2.1.2. User Interfaces

1. 로그인 화면: 이메일 또는 소셜로그인으로 로그인할 수 있다.
2. 대시보드: 학습자의 현재 학습 상태와 성과를 한눈에 확인할 수 있다. 완료된 문제, 피드백 받은 기록 등이 요약되어 있어 학습자가 자신의 진도를 쉽게 파악할 수 있다. 학습자는 자신의 학습 성과를 관리할 수 있으며, 자신의 강점, 개선이 필요한 부분을 볼 수 있다. 이를 통해 자기주도적인 학습 계획을 세울 수 있다.
3. 문제 목록 화면: 학습자 수준에 맞는 추천 문제들을 보여준다.
4. 학습 모드 및 챌린지 모드 선택: 학습 모드는 AI가 제공하는 단계별 지침과 피드백을 통해 사용자가 코딩 문제를 학습할 수 있도록

돕는다. 챌린지 모드는 사용자가 AI의 도움 없이 문제를 풀어볼 수 있는 실전 모드이다. 학습자는 자신이 배운 내용을 적용하고 문제 해결 능력을 테스트할 수 있다.

5. 문제 풀이 화면: 문제 설명, 요구 사항, 그리고 입력/출력 예제가 제공되는 화면이다. 문제 풀이를 위한 코드 입력 창이 있다.
  - a. (학습 모드) 피드백 패널: 학습자가 단계별 문제에 대한 코드를 입력하면 AI가 코드의 오류, 논리적 문제 등을 피드백한다. 시각화 피드백을 제공해 학습자가 코드 흐름을 직관적으로 이해하고 자신의 코드 오류나 알고리즘 로직을 쉽게 파악할 수 있도록 한다. 피드백 패널은 학습 모드일 때만 보여진다. 코드 옆에 위치하여 사용자가 문제를 쉽게 확인하고 코드를 수정할 수 있게 한다.
  - b. (챌린지 모드) 테스트 버튼: 예제를 실행해볼 수 있는 테스트 버튼이 있다. 테스트 결과는 하단에 출력되며, 오답일 경우 오답 유형과 에러 메시지가 표시된다.
    - i. 오답 유형: **Compile Error, Runtime Error, Wrong Answer, Time Limit Exceeded, Memory Limit Exceeded**
    - ii. 에러 메시지는 오답 유형이 **Compile Error**이거나 **Runtime Error**일 때 표시된다.
  - c. (챌린지 모드) 제출 버튼: 제출 버튼을 누르면 채점을 진행한다.
    - i. 오답일 경우 하단에 오답 유형과 에러 메시지가 표시된다.
      1. 에러 메시지는 오답 유형이 **Compile Error**이거나 **Runtime Error**일 때 표시된다.
    - ii. 정답일 경우 코드 제출 상세 화면으로 이동한다.
  - d. (챌린지 모드) 코드 제출 화면 이동 버튼: 코드 제출 화면으로 이동한다.
6. 코드 제출 화면: 코드 제출 이력을 볼 수 있다.
  - a. 각 제출한 코드 별로 제출 결과, 실행 시간(ms), 메모리 사용량(MB)을 테이블 형태로 보여준다.
    - i. 제출 결과: **Accepted, Compile Error, Runtime Error, Wrong Answer, Time Limit Exceeded, Memory Limit Exceeded**
  - b. 코드 제출 상세 화면: 각 테이블 행을 누르면 테이블에서 보여줬던 정보들과 더불어 제출한 소스 코드와 제출 결과에 따른 상세 정보들을 보여준다.

- i. 만약 제출 결과가 **Compile Error, Runtime Error**라면 에러 메시지를 보여준다.
- ii. 만약 제출 결과가 **Accepted**인 테이블 행을 클릭할 때 코드 분석 및 비교 결과를 보여준다.
  1. 시간복잡도/공간복잡도 분석해 빅오 표기법으로 보여주고, 개선할 수 있는 방안을 보여준다.
  2. 다른 사용자들의 코드와 비교해 제출된 답안 중 내 코드가 상위 몇 퍼센트의 속도/메모리인지 보여준다.

### 2.1.3. Hardware Interfaces

본 시스템은 웹 기반 애플리케이션으로 노트북이나 데스크탑 컴퓨터를 통해 사용할 수 있다. 최적의 사용 경험을 제공하기 위해 최소 사양으로 1.0GHz 이상의 프로세서가 있는 장치를 권장한다.

### 2.1.4. Software Interfaces

이 애플리케이션은 웹 기반이므로 Windows, macOS, Linux 등 여러 운영 체제에서 사용할 수 있다. 웹 브라우저에서 JavaScript가 활성화되어 있어야 하며, 최신 버전의 Chrome, Firefox, Edge와 같은 브라우저를 권장한다.

### 2.1.5. Communication Interfaces

"Step With Me"는 실시간 피드백을 제공하는 시스템이므로 안정적인 인터넷 연결이 필요하다. 이를 통해 AI 모델과 통신하여 사용자가 입력한 코드에 대해 즉각적으로 피드백을 줄 수 있다.

### 2.1.6. Memory Constraints

애플리케이션이 원활하게 실행되기 위해 최소 2GB의 RAM이 권장된다.

### 2.1.7. Operations

시스템 관리자

- 문제 데이터베이스 관리: 관리자는 학습자가 풀 문제를 데이터베이스에 추가, 수정 또는 삭제할 수 있다.
- 시스템 모니터링: 시스템 성능을 모니터링하여 피드백 속도, 서버 상태 등을 점검하고, 문제 발생 시 신속하게 대응하도록 한다.



- **AI 모델 업데이트 및 관리:** 최신 학습 데이터를 기반으로 **AI** 모델을 주기적으로 업데이트하여 피드백의 정확도와 학습 경험을 개선한다.

## 사용자

- **회원가입 및 로그인:** 학습자는 웹사이트에서 회원가입을 통해 계정을 생성하고, 로그인할 수 있다.
- **학습 모드 사용:** 학습 모드에서는 **AI**가 제공하는 단계별 학습과 실시간 피드백을 통해 학습자가 자신의 실력을 향상시킬 수 있도록 한다.
- **문제 풀이 및 피드백 받기:** 사용자는 문제를 풀고, **AI**가 제공하는 실시간 피드백을 통해 오류를 수정할 수 있다. 시각화 자료를 통해 코드 흐름을 직관적으로 이해할 수 있다.
- **챌린지 모드 사용:** 학습자는 **AI**의 도움 없이 문제를 해결해보는 챌린지 모드를 통해 실력을 테스트하고 자기주도적인 학습을 강화할 수 있다.
- **제출 코드 분석 결과 보기:** 챌린지 모드에서 제출된 답안의 시간 및 공간 복잡도와 최적화 방법을 알 수 있다. 다른 답안 코드들을 조회할 수 있고, 속도 및 메모리 비교 결과도 알 수 있다.
- **학습 기록 및 성과 조회:** 학습자는 자신의 진행상황 및 성과를 확인해 향후 학습 계획을 세울 수 있다.

## 2.2. Product Functions

### 2.2.1. 회원가입 및 로그인

개인화된 학습 경험을 제공하기 위해서는 사용자 계정 생성 및 로그인 필수 기능이다. 새로운 사용자가 시스템을 이용하려면 회원가입을 통해 계정을 생성해야 한다. 회원가입 시 필요한 정보로는 사용자 이름, 이메일 주소, 비밀번호가 있다. 사용자는 등록한 이메일과 비밀번호를 통해 로그인할 수 있다. 로그인 후 사용자는 자신의 학습 진도, 이전에 해결한 문제, 피드백 기록 등을 확인할 수 있다. 이를 통해 개별 학습 진행 상황에 맞춘 경험을 제공한다.

### 2.2.2. 맞춤형 문제 제공

사용자의 학습 수준과 이력을 바탕으로 맞춤형 문제를 제공한다. 데이터베이스에 저장된 학습자의 목표, 이전 문제 풀이 기록, 오답 패턴, 피드백을 분석해 학습자의 강점과 약점을 기반으로 다음 학습 단계에

필요한 문제를 제공한다. 예를 들어, 자주 틀리는 특정 알고리즘을 반복 학습하거나 새로운 유형의 문제에 도전할 수 있도록 한다.

### **2.2.3. 단계별 학습 및 실시간 피드백 (학습 모드)**

사용자가 단계별로 문제를 풀고 코드를 제출하면 AI 모델이 실시간으로 피드백을 제공한다. 코드 오류, 논리적 오류 등에 대한 피드백이 제공된다. 시각화 도구를 활용해 코드 실행과정을 시각적으로 보여준다.

### **2.2.4. 코드 분석 및 비교 (챌린지 모드)**

시간 복잡도와 공간 복잡도를 분석하여 보다 최적화된 솔루션을 제안하거나 개선할 수 있는 부분을 사용자에게 안내한다. 제출된 답안 중 사용자의 코드가 상위 몇 퍼센트의 속도/메모리인지 비교해 보여준다.

### **2.2.5. 학습 기록 및 성과 관리**

사용자의 학습 진행도, 피드백 기록, 성공한 문제 등을 데이터베이스에 저장하여 관리한다. 사용자는 대시보드에서 학습 이력과 성과를 조회할 수 있다. 사용자의 강점 및 성과 분석을 통해 배지, 업적 등의 아이템을 제공한다.

### **2.2.6. 학습 목표 설정**

사용자는 학습 목표(예: 알고리즘 이해, 문제 해결 능력 향상 등)를 설정할 수 있으며, 목표에 맞는 문제 유형과 학습 경로를 추천받는다.

## **2.3. User Classes and Characteristics**

### **2.3.1. 초급 학습자 (Beginner Learners)**

코딩과 프로그래밍에 대한 사전 지식이 거의 없거나 처음 접하는 사용자들이다. 기본적인 알고리즘 개념이나 코딩 문법에 익숙하지 않을 수 있다. 단계별 학습, 기본적인 개념 설명, 쉬운 문제 제공, 실시간 피드백 등 기초 수준의 학습 지원이 필요하다.

### **2.3.2. 중급 학습자 (Intermediate Learners)**

기초적인 프로그래밍 개념과 문법은 이해하고 있으며, 간단한 문제는 스스로 해결할 수 있다. 더 복잡한 알고리즘과 자료구조를 배우고 싶어 한다. 다양한 문제 유형, 코드 분석 및 최적화 방법에 대한 피드백을 필요로 한다.

### 2.3.3. 고급 학습자 (Advanced Learners)

복잡한 문제 해결과 최적화에 관심이 많고, 다양한 알고리즘과 데이터 구조를 다뤄본 경험이 있다. 고난이도 문제 제공, 다른 고급 사용자와의 코드 비교, 최적화 피드백을 필요로 한다.

## 2.4. Design and Implementation Constraints

- 모든 요청은 정해진 시간 내에 응답을 받도록 보장하여 시간적 성능을 보장해야 한다. 또한 실행시 메모리 제한을 넘지 않도록 하여 메모리 성능을 보장해야 한다.
- 사용자의 개인정보와 관련된 데이터는 암호화하여 보호되어야 한다.
- 최신 플랫폼 표준 규격을 준수해야 한다.
- 개발환경에서의 하드웨어 사항 내에서 개발을 진행하여야 한다.
- 소프트웨어 사용에 대한 규제, 개인정보 활용에 대한 규제 등 소프트웨어 개발시 적용되는 규제를 준수하여야 한다.

## 2.5. Assumptions and Dependencies

“Step With Me”는 코드를 입력하며 프로그래밍을 학습하는 프로그램이기에, 사용자는 코드를 입력할 수 있는 입력 수단이 존재하는 환경에서 실행한다고 가정한다. 또한 웹 기반 서비스이기 때문에 사용자의 디바이스가 인터넷에 연결된 환경이라고 가정한다. LLM 기반의 인터랙션을 위해 클라이언트 장치의 연산 능력이 최소한으로 요구되며, 저사양 장치에서는 최적의 사용자 경험 제공이 어려울 수 있기에 이러한 조건을 만족시킨 환경이라고 가정한다. 이러한 조건이 충족되지 않는 환경에서는 시스템이 원활히 작동하지 않을 수 있다.

# 3 External Interface Requirements

## 3.1 External interfaces

### 3.1.1 User interface

1. 마우스 및 키보드를 통한 입력 처리
  - a. 목적: 서비스 사용자가 키보드 및 마우스 입력을 통해 시스템에 명령 전달
  - b. 예시: 로그인 정보 입력, 회원가입 정보 입력, 알고리즘 문제 선택, 챌린지 모드/학습 모드 선택, 알고리즘 코드 입력, 문제 제출 등
  - c. 입력 소스/출력 대상: 사용자/컴퓨터 기기
  - d. 유효 범위: 화면에서의 텍스트 입력창 또는 버튼의 개수에 따름
  - e. 정확도: 사용자의 마우스 및 키보드 입력에 따름
  - f. 측정 단위: 버튼 클릭/키보드 입력
  - g. 데이터 형식: 문자열/마우스 이벤트

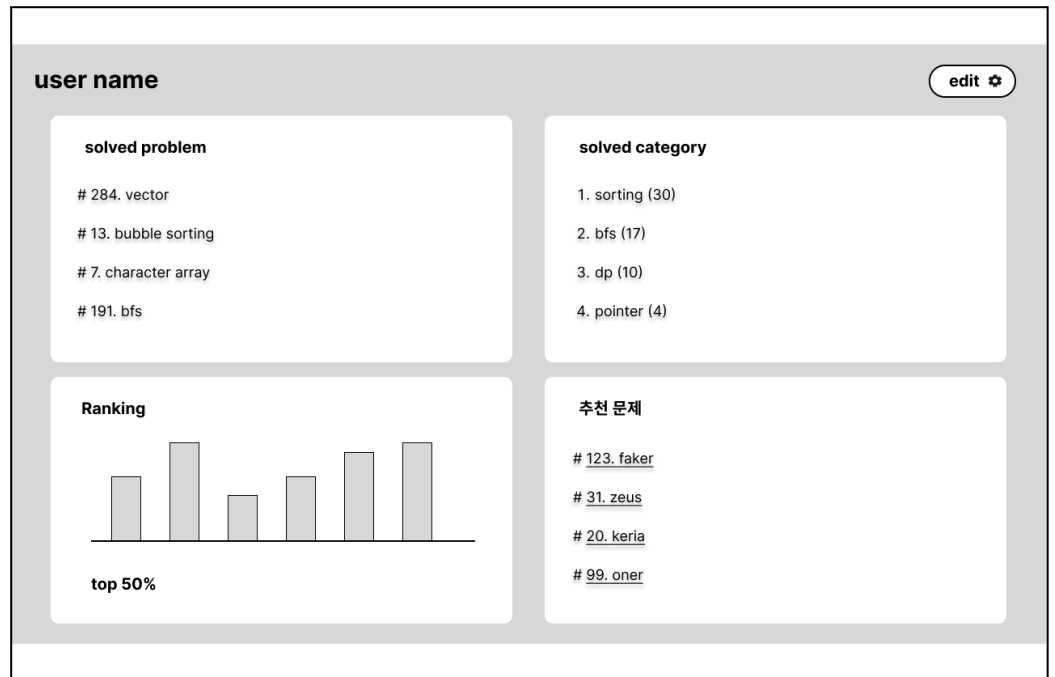
- h. 타이밍: 비정기적인 사용자의 입력에 대해 즉각적인 명령 수행
- i. 다른 입력/출력과 관계: 유효하지 않은 입력 시 경고 창 발생 및 추가적인 확인 요구

## 2. 화면 출력

- a. 목적: 서비스 사용자에게 제공하는 인터페이스
- b. 예시: 로그인 화면, 대시보드 화면, 문제 풀이 화면, 코드 제출 화면 등 서비스 이용 전반
- c. 입력 소스/출력 대상: 웹 서버/사용자
- d. 유효 범위: 서비스 이용 전반
- e. 정확도: 유저의 모니터 출력 정확도와 디스플레이 해상도, 네트워크 환경에 따름
- f. 측정 단위: 모니터 화면
- g. 데이터 형식: 이미지, 텍스트, html, javascript
- h. 타이밍: 즉각적
- i. 다른 입력/출력과 관계: 사용자의 입력에 따른 화면 전환
- j. 화면 형식 및 구성:
  - i. 로그인 화면: 이메일/비밀번호 입력창, 로그인 버튼, 소셜 로그인 버튼, 회원가입, 비밀번호 찾기 버튼

The diagram shows a login interface within a rounded rectangle. At the top, the word "Login" is centered. Below it, the "Email" label is to the left of a text input field. The "Password" label is to the left of another text input field. To the right of the "Email" field is a button labeled "login". Below the "Email" and "Password" fields, there are three buttons: "forgot password", "sign up", and "social login".

- ii. 대시보드: 학습자가 최근 완료한 문제, 자주 푼 유형, 랭킹 등의 학습 요약, 문제 목록 버튼, 프로필 수정 버튼, 문제 목록 버튼



- iii. 문제 목록 화면: 추천 문제들의 난이도, 유형, 정답률 등이 표시된 리스트 표시

Recommended Problems				
No.	이름	유형	난이도	정답률
12	<a href="#">problem A &gt;</a>	bfs		60%
30	<a href="#">problem B &gt;</a>	bfs		42%
82	<a href="#">problem C &gt;</a>	bfs		15%
5	<a href="#">problem D &gt;</a>	sorting		35%
392	<a href="#">problem E &gt;</a>	pointer		59%
245	<a href="#">problem F &gt;</a>	dfs		20%
59	<a href="#">problem G &gt;</a>	pointer		70%

- iv. 모드 선택 화면: 챌린지 모드와 학습 모드 버튼. 각 버튼 아래에는 모드에 대한 설명 표시

### Select Mode

#### Study Mode

- 문제의 로직을 이해하며 풀어보기
- 문제를 여러 부분으로 나눠서 풀어보기
- LLM 튜터와 함께 즉각적 피드백

#### Challenge Mode

- 공부한 프로그래밍을 바탕으로 실전 문제 풀어보기
- 풀이중 피드백은 제공되지 않음
- 채점 및 다양한 피드백 제공

- v. 문제 풀이 화면: 문제에 대한 단계별 가이드라인(학습 모드), 코드 입력란, 코드 제출/실행 버튼, 실행 결과창, 피드백 발생 시 피드백 패널(학습 모드), 나가기 버튼

# 123. Find max

quit

주어진 배열에서 가장 큰 값을 찾으시오.

입력값 : 임의의 숫자 배열  
출력값 : 배열 내의 숫자 중 가장 큰 값

배열 내의 각 숫자 N :  
0 < N < 10000

예시 :  
입력값 : [2,7,1,60,70,100,24]  
출력값 : 100

조건 :  
time limit : 10ms  
space limit : 1MB

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

submit

1 int max = arr[-1];

2

3

Step 2. 첫 항부터 끝 항까지 반복하며 최댓값을 업데이트해요

Step 3. 최댓값을 반환해요

# 123. Find max

주어진 배열에서 가장 큰 값을 찾으시오.

입력값 : 임의의 숫자 배열  
출력값 : 배열 내의 숫자 중 가장 큰 값

배열 내의 각 숫자 N :  
0 < N < 10000

예시 :  
입력값 : [2,7,1,60,70,100,24]  
출력값 : 100

조건 :  
time limit : 10ms  
space limit : 1MB

Step 1. 최댓값을 담은 변수를 선언하고 배열의

1 int max = arr[-1];

2

3

Step 2. 첫 항부터 끝 항까지 반복하며 최댓값을

Step 3. 최댓값을 반환해요

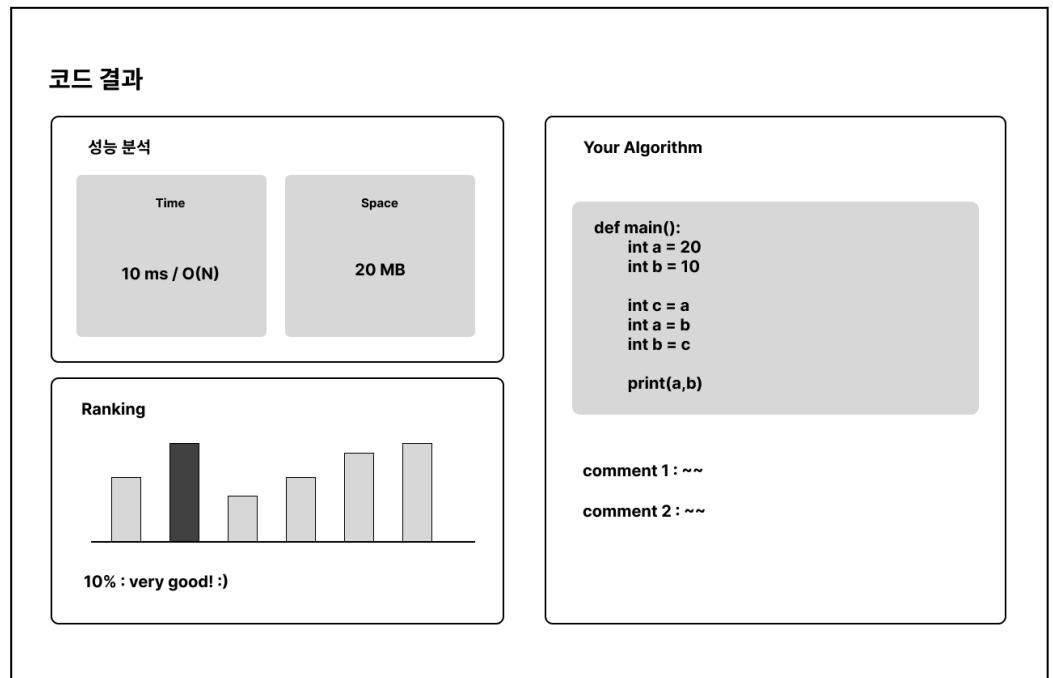
> 실행 결과  
error message:  
~~

segmentic 에러가 발생했어요.  
index의 범위에 대해 생각해 보세요

> 시각화 피드백



- vi. 코드 제출 화면: 메모리 사용량/속도 측면 사용자의 풀이 분석, 순위(또는 백분위) 표시



### 3.1.2 Hardware Interface

#### 1. CPU 및 RAM

- 목적: 원활한 시스템 운영을 위한 최소 성능 기준 제공
- 입력 소스/출력 대상: CPU 및 RAM / 사용자 컴퓨터
- 유효 범위/정확도/허용 오차: 요구되는 최소 사양을 만족해야 최적 성능 발휘 (2.4.4에 명시됨)
- 측정 단위: GHz (CPU), GB (RAM)
- 타이밍: 지속적
- 다른 입력/출력과 관계: 메모리 제한 미충족 시 느린 반응 속도 가능성, 사용자 경험 저하

#### 2. 네트워크 연결

- 목적: 실시간 데이터 통신 및 피드백 제공
- 입력 소스/출력 대상: 인터넷 서비스 / 웹 애플리케이션
- 유효 범위/정확도/허용 오차: 최소 10Mbps 인터넷 속도 권장
- 측정 단위: Mbps
- 타이밍: 지속적
- 다른 입력/출력과 관계: 네트워크 오류 시 지연 발생 가능, 피드백 속도에 영향

### 3.1.3 Software Interface

#### 1. 웹 브라우저 호환성

- a. 목적: 다양한 OS에서 동일한 사용자 경험을 제공하기 위해 최신 브라우저 호환성 제공
- b. 입력 소스/출력 대상: 웹 애플리케이션 / 사용자 브라우저
- c. 유효 범위/정확도/허용 오차: 최신 버전의 Chrome, Firefox, Safari, Edge 권장
- d. 측정 단위: 브라우저 버전
- e. 타이밍: 사용자 요청 시
- f. 다른 입력/출력과 관계: Internet Explorer 등 구형 브라우저에서는 비정상 작동 가능성 있음

#### 2. 백엔드 API

- a. 목적: 사용자 인증, 학습 데이터, 코드 평가를 클라이언트와 서버 간 통신을 통해 제공
- b. 입력 소스/출력 대상: 사용자 및 서버 / 클라이언트 애플리케이션
- c. 유효 범위/정확도/허용 오차: 정확한 데이터 제공, 호출 실패 시 오류 메시지 반환
- d. 측정 단위: API 호출
- e. 타이밍: 실시간
- f. 다른 입력/출력과 관계: 사용자 요청에 따른 데이터 전송 및 피드백 제공

#### 3. 데이터베이스 통신

- a. 목적: 학습자의 진행 상태, 문제 풀이 기록, 피드백 내역 등 학습 데이터를 저장 및 조회하여 맞춤형 학습 경험 제공
- b. 입력 소스/출력 대상: 웹 애플리케이션 / PostgreSQL 데이터베이스
- c. 유효 범위/정확도/허용 오차: 데이터의 무결성을 유지하며, 저장 및 조회 작업의 성공 여부는 시스템에 의해 확인
- d. 측정 단위: 데이터 트랜잭션 수
- e. 타이밍: 실시간 또는 요청 시

- f. 다른 입력/출력과 관계: 사용자 활동에 따라 데이터베이스가 업데이트되며, 학습 진행 상황 조회 시 데이터베이스로부터 데이터를 불러옴
- g. **Database** 사용 기술 및 구조:
  - i. 데이터베이스 관리 시스템 (DBMS): PostgreSQL
  - ii. 트랜잭션 처리: 데이터의 일관성을 보장하기 위해 트랜잭션 단위로 데이터 작업 처리
  - iii. 주요 데이터 항목:
    - 1. 사용자 정보: 사용자 계정, 로그인 정보, 학습 목표 및 성취 기록
    - 2. 문제 풀이 기록: 문제 ID, 풀이 코드, 제출 시간, 성공 여부
    - 3. 피드백 기록: 피드백 내용, 코드 오류 및 논리적 문제 정보, 개선 사항
    - 4. 시스템 설정: 문제 난이도, 학습 모드 및 챌린지 모드 설정, 사용자 맞춤형 추천 설정

#### 4. LLM 인터페이스

- a. 목적: 실시간 AI 피드백을 위해 LLM 모델과의 데이터 통신
- b. 입력 소스/출력 대상: 사용자 코드 / LLM 서버
- c. 유효 범위/정확도/허용 오차: 응답 지연 시 사용자에게 알림 표시
- d. 측정 단위: gRPC 호출 수
- e. 타이밍: 사용자 요청에 따른 실시간 응답
- f. 다른 입/출력과 관계: 서버 응답 지연 발생 시 시스템 지연 발생

### 3.1.4 Communication Interface

#### 1. HTTPS

- a. 목적: 웹 애플리케이션과 서버 간의 데이터 전송 및 보안을 위해 HTTPS 사용
- b. 입력 소스/출력 대상: 클라이언트 / 서버
- c. 유효 범위/정확도/허용 오차: SSL을 통한 암호화된 데이터 전송
- d. 측정 단위: HTTPS 요청

- e. 타이밍: 지속적
- f. 다른 입/출력과 관계: 인증되지 않은 연결 시 데이터 전송 불가

## 2. WebSocket

- a. 목적: 실시간 피드백 제공을 위한 클라이언트와 서버 간 지속적 데이터 전송
- b. 입력 소스/출력 대상: 클라이언트 / 서버
- c. 유효 범위/정확도/허용 오차: 네트워크 상태에 따라 성능 변화, 실시간 데이터 전송
- d. 측정 단위: **WebSocket** 메시지 수
- e. 타이밍: 지속적
- f. 다른 입/출력과 관계: 실시간 통신 지연 시 시스템 지연 발생 가능

## 3. gRPC 프로토콜

- a. 목적: AI 모델 서버와의 효율적인 데이터 처리와 전송
- b. 입력 소스/출력 대상: 사용자 코드 / **LLM** 서버
- c. 유효 범위/정확도/허용 오차: 데이터 처리 속도 유지, **JSON** 데이터 전송
- d. 측정 단위: **gRPC** 호출 수
- e. 타이밍: 사용자 요청 시
- f. 다른 입/출력과 관계: **API** 호출 실패 시 대체 피드백 제공

## 3.2 Function requirements

### 3.2.1 Use Case

use case name	Register
<i>Actor</i>	회원이입하지 않은 사용자
<i>Description</i>	회원이입하지 않은 사용자가 서비스 계정을 생성하는 과정

<b>Normal Course</b>	<ol style="list-style-type: none"> <li>1. 사용자가 웹사이트에 접속하면 로그인 페이지가 나타난다.</li> <li>2. 이메일로 회원가입을 원하면 회원가입 버튼을, 소셜 회원가입을 원하면 원하는 소셜 아이콘을 클릭한다.</li> <li>3. (이메일 회원가입 시) 사용자는 이메일을 작성하여 인증 버튼을 누른다. 이메일로 발송된 인증 코드를 입력한 후 비밀번호와 비밀번호 확인을 입력한다. 이메일 인증과 비밀번호가 정해진 기준을 만족하며, 비밀번호 확인과 일치하면 등록이 완료되고 메인화면으로 돌아간다.과 일치하면 등록이 완료되고 메인화면으로 돌아간다.</li> <li>4. (소셜 회원가입 시) 사용자는 가입을 원하는 소셜 계정을 선택하고 개인정보 수집 동의를 하면 등록이 완료되며 메인화면으로 돌아간다.</li> </ol>
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. 사용자의 이메일(또는 소셜 계정)은 아직 서비스에 회원가입되어있지 않다.</li> <li>2. 사용자는 유효한 이메일(또는 소셜 계정)과 비밀번호를 입력하고, 이메일 인증을 완료했다.</li> </ol>
<b>Post Condition</b>	<ol style="list-style-type: none"> <li>1. 사용자의 이메일과 비밀번호는 데이터베이스에 저장된다.</li> <li>2. 비밀번호는 암호화되어 저장된다.</li> <li>3. 사용자의 계정 정보로 바로 로그인이 가능하다.</li> </ol>
<b>Assumptions</b>	
<b>use case name</b>	<b>Login/Logout</b>
<b>Actor</b>	회원가입한 사용자
<b>Description</b>	회원가입한 사용자가 서비스에 로그인/로그아웃을 시도하는 과정
<b>Normal Course</b>	<ol style="list-style-type: none"> <li>1. 사용자가 웹사이트에 접속하면 로그인 페이지가 나타난다.</li> <li>2. (이메일 로그인 시) 사용자는 이메일과 비밀번호를 입력하고 로그인 버튼을 클릭한다.</li> <li>3. (소셜 로그인 시) 사용자는 원하는 로그인 버튼을 클릭한다. 사용자는 가입된 소셜 계정을 선택하고 로그인 요청을 승인한다.</li> <li>4. 인증에 성공하면 메인 화면으로 이동한다.</li> <li>5. 로그아웃 버튼을 누르면 현재 세션이 사라지고 로그인 페이지로 돌아온다.</li> </ol>
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. 사용자는 이미 회원가입을 완료하여 계정을 보유하고 있어야 한다.</li> <li>2. 사용자는 현재 로그인되어있지 않다.</li> <li>3. 사용자가 입력한 이메일과 비밀번호가 유효하다.</li> <li>4. 사용자의 계정이 유효하다.</li> </ol>
<b>Post Condition</b>	<ol style="list-style-type: none"> <li>1. 사용자는 메인 화면에 로그인한 상태로 접속한다.</li> <li>2. 사용자 세션이 시작되며 이후 http요청마다 유저 정보가</li> </ol>

	인증된다.
<i>Assumptions</i>	사용자의 네트워크가 안정적이어야한다.
<b>use case name</b>	<b>Select algorithm problem</b>
<i>Actor</i>	로그인 한 사용자
<i>Description</i>	사용자가 풀고자 하는 알고리즘 문제를 선택하는 과정
<i>Normal Course</i>	<ol style="list-style-type: none"> <li>1. 사용자는 문제 목록 화면에 입장한다.</li> <li>2. 원하는 문제 유형과 난이도에 대해 적절한 문제를 선택하고, 문제 이름을 클릭한다.</li> </ol>
<i>Precondition</i>	<ol style="list-style-type: none"> <li>1. 사용자는 로그인되어 있다.</li> <li>2. 사용자는 문제 목록 화면에 있다.</li> <li>3. 사용자에게 문제 목록들이 주어져서 리스트로 표시되어 있다.</li> </ol>
<i>Post Condition</i>	사용자는 모드 선택 화면으로 이동한다.
<i>Assumptions</i>	사용자의 네트워크가 안정적이어야한다.

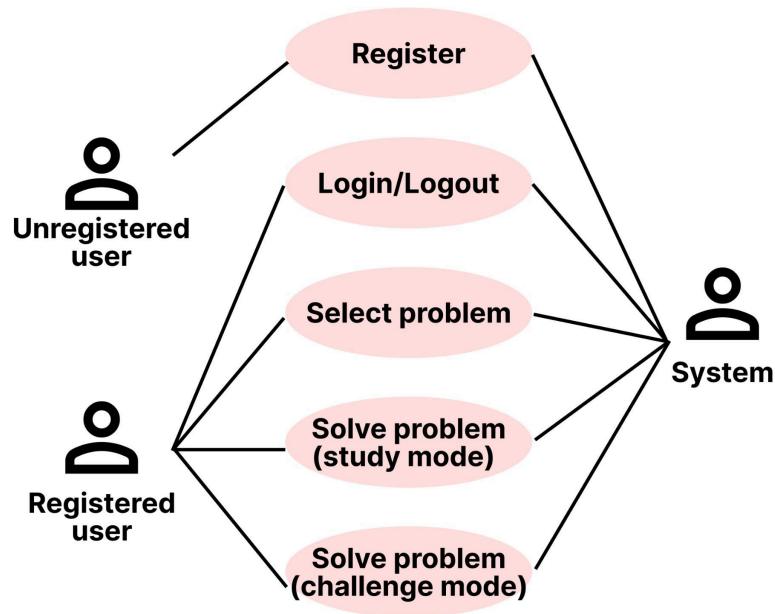
<b>use case name</b>	<b>Solve problem (Study mode)</b>
<i>Actor</i>	로그인 한 사용자
<i>Description</i>	사용자가 스터디 모드에서 문제를 풀이해나가는 과정
<i>Normal Course</i>	<ol style="list-style-type: none"> <li>1. 사용자가 선택한 문제의 페이지에 접속한다.</li> <li>2. 문제에 대한 자세한 설명을 통해 문제를 파악한다.</li> <li>3. 문제 옆에 해결을 위한 <b>logic</b>을 단계별로 구별해 둔 <b>block</b>이 존재하며, 해당 <b>block</b>엔 <b>logic</b> 가이드라인과 코드를 작성할 수 있는 공간과 입력 버튼이 존재한다.</li> <li>4. 사용자는 해당 <b>logic block</b>에서 가이드라인을 읽고 코드를 작성한다.</li> <li>5. 입력이 완료되면 입력 버튼을 눌러 코드를 제출한다.</li> <li>6. 제출하면 해당 부분에 대한 채점과 피드백 결과가 제공된다.</li> <li>7. 정답일 경우 다음 <b>logic block</b>으로 넘어갈 수 있고, 오답일 경우 피드백을 제공받고 다시 해당 <b>block</b> 풀이를 시도한다.</li> </ol>
<i>Precondition</i>	<ol style="list-style-type: none"> <li>1. 사용자는 로그인되어있고, 문제 풀이 화면에 있다.</li> <li>2. 문제 풀이 모드가 학습 모드로 설정되어 있다.</li> <li>3. 각 단계별 <b>logic block</b>이 준비되어 있다.</li> <li>4. 각 <b>logic block</b>에는 코드 입력란과 채점 버튼이 활성화되어 있다.</li> </ol>

<i>Post Condition</i>	<ol style="list-style-type: none"> <li>1. 사용자는 각 <b>logic block</b>에 대해 채점 및 피드백을 받는다.</li> <li>2. 정답을 제출한 <b>logic block</b>은 완료 상태로 변경되고, 입력이 비활성화된다.</li> <li>3. 문제가 전부 해결되면 결과 화면으로 이동한다.</li> <li>4. 사용자가 해결한 문제 정보, 피드백 정보들이 데이터베이스에 저장되며, 이후 문제 추천과 학습 기록에 반영된다.</li> </ol>
<i>Assumptions</i>	사용자의 네트워크가 안정적이어야한다.

<b>use case name</b>	<b>Solve problem (Challenge mode)</b>
<i>Actor</i>	로그인 한 사용자
<i>Description</i>	사용자가 챌린지 모드에서 문제를 풀이해나가는 과정
<i>Normal Course</i>	<ol style="list-style-type: none"> <li>1. 사용자가 선택한 문제의 페이지에 접속한다.</li> <li>2. 문제에 대한 자세한 설명을 통해 문제를 파악한다.</li> <li>3. 문제 옆엔 코드를 작성할 수 있는 공간과 입력 버튼이 존재한다.</li> <li>4. 사용자는 문제를 풀기 위한 코드를 작성하고 입력버튼을 눌러 코드를 제출한다.</li> <li>5. 제출하면 채점결과가 제공된다.</li> <li>6. 정답일 경우 제출한 코드에 대한 피드백이 제공되며 오답일 경우 에러 메세지 등의 에러 정보만 제공되며 피드백이 제공되지 않은 채로 다시 해당 문제 풀이를 시도한다.</li> </ol>
<i>Precondition</i>	<ol style="list-style-type: none"> <li>1. 사용자는 로그인되어있고, 문제 풀이 화면에 있다.</li> <li>2. 문제 풀이 모드가 챌린지 모드로 설정되어 있다.</li> <li>3. 문제 채점 버튼이 활성화되어 있다.</li> </ol>
<i>Post Condition</i>	<ol style="list-style-type: none"> <li>1. 사용자는 전체 문제 풀이에 대해 채점 및 피드백을 받는다.</li> <li>2. 정답 시 결과 페이지로 이동하고, 오답 시 코드를 수정 후 재채점할 수 있다.</li> <li>3. 사용자가 해결한 문제 정보, 피드백 정보들이 데이터베이스에 저장되며, 이후 문제 추천과 학습 기록에 반영된다.</li> </ol>
<i>Assumptions</i>	사용자의 네트워크가 안정적이어야한다.

### 3.2.2 Use Case Diagram



### 3.3 Performance requirements

#### 3.3.1 Static Numerical Requirements

- 이 애플리케이션은 다양한 운영체제에서 최신 웹 브라우저를 통해 접속할 수 있으며, 각 운영체제당 최소 1,000명의 동시 사용자를 지원하도록 설계한다.
- 시스템은 평균적으로 최대 10,000명의 동시 사용자가 원활하게 사용할 수 있도록 설계되어야 하며, 트래픽 피크 시 최대 15,000명까지 지원해야 한다.

#### 3.3.2 Dynamic Numerical Requirements

- 시스템은 초당 최대 500개의 데이터베이스 트랜잭션(학습 데이터 저장, 문제 풀이 결과 저장 등)을 처리할 수 있어야 한다.



- 사용자 코드 제출 후 피드백은 평균 1초 내로 제공되어야 하며, LLM 서버에서의 응답은 최대 2초를 초과하지 않아야 한다.

## 3.4 Logical database requirements

본 시스템의 데이터베이스는 학습자의 진행 상태, 문제 풀이 기록, 피드백 내역 등을 저장하고 조회하여 맞춤형 학습 경험을 제공한다. 각 데이터 유형에 대해 정보의 중요도, 사용 빈도, 접근 권한, 무결성 제약을 정의하여 데이터베이스의 일관성과 신뢰성을 확보한다.

### 3.4.1 사용자 정보

- 정보 유형: 사용자 ID, 이름, 이메일, 암호화된 비밀번호, 학습 목표 및 성취 기록
- 사용 빈도: 상시 조회 및 사용
- 접근 권한: 시스템 관리자, 인증된 사용자 자신
- 무결성 제약:
  - 이메일은 중복을 허용하지 않는다.
  - 암호화된 비밀번호는 복호화가 불가능한 형태로 저장되어야 한다.
  - 이메일 형식은 표준 이메일 형식을 준수해야 하며, 비밀번호는 특정 강도 요건을 만족해야 한다(예: 최소 8자, 숫자 포함 등).

### 3.4.2 문제 정보

- 정보 유형: 문제 ID, 제목, 설명, 입력 예시, 출력 예시, 난이도, 문제 유형(예: 정렬, 탐색 등), 시간 제한(ms), 메모리 제한(MB), 단계 별 하위 문제, 채점용 테스트케이스
- 사용 빈도: 높은 빈도로 조회, 주기적인 업데이트
- 접근 권한: 시스템 관리자(생성, 수정, 삭제), 사용자(조회)
- 무결성 제약:
  - 문제 ID는 고유해야 하며, 난이도는 정해진 레벨(초급, 중급, 고급 등)에 따라야 한다.
  - 문제 유형은 사전에 정의된 카테고리에 맞아야 한다.
  - 문제의 설명과 예시는 각 문제와 일관된 요구사항을 만족하도록 검증되어야 한다.

### 3.4.3 문제 풀이 기록

- 정보 유형: 사용자 ID, 문제 ID, 제출 시간, 제출된 코드, 풀이 결과(성공/실패), 사용된 시간/메모리
- 사용 빈도: 빈번한 쓰기 및 조회(피드백 및 성취도 분석에 사용)
- 접근 권한: 인증된 사용자 자신(조회), 시스템 관리자(조회, 분석용 접근)
- 무결성 제약:
  - 사용자 ID와 문제 ID의 조합은 유일해야 하며, 각 제출 시 새로운 레코드로 기록되어야 한다.
  - 성공 여부는 특정 기준(예: 모든 테스트 케이스 통과 여부)에 따라 정확하게 기록되어야 한다.

#### 3.4.4 피드백 데이터

- 정보 유형: 피드백 ID, 사용자 ID, 문제 ID, 코드 오류 정보, 논리 오류, 개선 제안, 시각적 피드백 데이터
- 사용 빈도: 빈번한 조회(학습 모드 피드백 제공), 쓰기는 문제 풀이 시 자동 발생
- 접근 권한: 인증된 사용자(자신의 피드백 조회), 시스템 관리자(조회 및 데이터 검토)
- 무결성 제약:
  - 피드백 ID는 고유해야 하며, 오류 정보와 개선 제안은 정확하고 유효한 내용만 포함되어야 한다.
  - 피드백은 제출된 코드에 일관된 방식으로 연결되어야 하며, 코드 오류 유형과 시각적 피드백은 사전 정의된 형식에 맞춰 저장된다.

#### 3.4.5 추천 데이터

- 정보 유형: 사용자 ID, 추천 문제 목록(문제 ID들), 추천 근거(예: 오답 경향성, 특정 유형의 출제 빈도 등)
- 사용 빈도: 주기적 조회(대시보드 추천 문제 표시), 주기적 업데이트(사용자 활동에 따라 추천 갱신)
- 접근 권한: 시스템 관리자(조회 및 수정), 인증된 사용자 자신(조회)
- 무결성 제약:
  - 추천 문제 목록은 사용자의 학습 이력과 일관되게 생성되어야 하며, 동일 문제의 중복 추천을 방지해야 한다.
  - 각 문제의 난이도는 사용자의 수준에 적합해야 하며, 문제의 설명과 예시는 해당 추천 목적에 적합해야 한다.

### 3.4.6 타 사용자와의 비교 데이터

- 정보 유형: 사용자 ID, 문제 ID, 풀이 시간 및 메모리 사용량 비교 결과(백분위)
- 사용 빈도: 빈번한 조회(챌린지 모드 분석 결과 제공)
- 접근 권한: 인증된 사용자(자신의 성과와 다른 사용자의 결과 비교 조회)
- 무결성 제약:
  - 사용자 성과 기록은 일관된 기준으로 비교되어야 하며, 백분위 계산은 정확해야 한다.
  - 타 사용자의 결과는 개인정보 보호 원칙에 따라 익명화되어야 하며, 각 사용자별 개별 비교가 아닌 통계적 정보로 제공되어야 한다.

## 3.5 Design constraints

### 3.5 Design Constraints

#### 3.5.1 성능 제약사항

- 백엔드 서버의 응답은 1초 이내, LLM 모델 서버의 응답은 2초 이내로 제한한다.
- LLM 호출 시 메모리 사용량은 4GB를 초과하지 않도록 한다.

#### 3.5.2 하드웨어 제약사항

- 사용자는 최소 1.0GHz CPU와 4GB RAM을 갖춘 장치에서 애플리케이션을 원활히 사용할 수 있어야 하며, 서버는 Linux OS에서 NVIDIA GPU 지원과 함께 효율적으로 작동한다.

#### 3.5.3 플랫폼 제약사항

- 최신 웹 표준을 준수하는 브라우저(Chrome, Firefox, Safari, Edge)를 지원하며, Internet Explorer는 지원하지 않는다.

#### 3.5.4 규제 정책

- 모든 사용자 데이터는 SSL 암호화를 통해 전송하며, 개인정보보호법을 준수한다. LLM 및 외부 소프트웨어 구성 요소는 관련 저작권 및 라이선스를 준수하여 사용한다.

## 3.6 Software system attributes

### 3.6.1 Reliability

- 사용자 로그인, 학습 모드 및 챌린지 모드, 피드백 제공 기능은 시스템 중단 없이 안정적으로 작동하도록 한다.
- 데이터베이스에 저장된 사용자 기록(문제 풀이 기록, 피드백)은 매일 백업하며, 데이터 손실 시 빠르게 복구할 수 있도록 리던던시(redundancy)를 유지한다.

### 3.6.2 Availability

- 시스템 가용성은 연중무휴 **24/7, 99.5%** 이상을 목표로 한다.
- 주요 서비스 장애 시 자동화된 알림 및 재시작 기능을 갖추며, 필요 시 시스템 관리자가 신속히 복구할 수 있도록 한다.

### 3.6.3 Security

- 모든 사용자 데이터는 **SSL** 암호화를 통해 전송하고, 비밀번호는 해시 처리된 상태로 저장한다.
- 사용자 접근 로그를 기록하여 이상 접근 발생 시 알림을 전송하며, 주요 시스템 이벤트와 오류를 모니터링하여 시스템 보안을 강화한다.
- 데이터 변경 및 전송 시 무결성을 유지하도록 검증하며, 접근 제어를 통해 권한이 없는 사용자의 접근을 차단한다.

### 3.6.4 Maintainability

- 모듈화된 구조로 설계하여 특정 기능(예: 문제 추천 엔진, 피드백 엔진)이 독립적으로 유지보수 가능하도록 한다.
- 공통 모듈을 다른 기능에서 재사용할 수 있도록 하며, 모듈 간 의존성을 최소화하여 유지보수를 용이하게 한다.
- 외부 시스템과의 통신은 표준화된 **API**를 통해 수행하며, 유지보수와 업데이트가 수월하도록 문서화를 철저히 한다.

### 3.6.5 Portability

- Windows, macOS, Linux 등 여러 운영체제에서 최신 웹 브라우저를 통해 접근 가능하도록 한다.

- 마이크로서비스 아키텍처 기반으로 설계하여 개별 서비스가 독립적으로 운영되도록 하며, 다른 호스트 환경으로의 이식이 용이하도록 호스트 의존적인 코드를 최소화한다.