

# System Software Design

Project Draft

202355550

Kang-Kyung Min

# Quiz generator

## [Purpose]

- To support students' learning by generating quizzes and provide instant feedback
- To make learning more fun by adding game-like elements

### [Functional Requirements]

- Use OpenAI API to generate quizzes
- Get user text for the topic of the quiz  
(If it turns out to be an overscope the topic choice may be limited)
- Allow users to specify quiz types and difficulty

Topic	System Design
Context	(Insert a long text about OSI here)
Difficulty	Easy / Medium / Hard
Quiz Type	Multiple choice / Fill in the blanks

-The AI response should be formatted so it can be parsed:

example for the multiple choice quiz type)

Question: What's the name of 6th OSI Layer?

Choices:

1. Presentation Layer
2. Application Layer
3. Transport Layer
4. Session Layer

1	2	3	4
---	---	---	---

Answer: 1

- Generate the UI from the AI response and get the user input
- Compare the user input and the answer, then show the result
- Store the result in a database for future reviews

Date	Result	Topic	
2024/5/10	5/8	C++	Review ->
2024/5/16	7/8	C++	Review ->
2024/5/17	3/6	System Design	Review ->

### [Non-Functional Requirements]

- Scalability: Suitable for small number of students (not for big companies)
- Accuracy: Wrong answers/choices by the AI should be minimized
- Response time: waiting time between each part of the program shouldn't be too long

## [Components]

### Load Balancer

- Handle the traffic efficiently
- Nginx

### Servers

- Implement the functional/non-functional requirements
- Node.js

### API

- Generate quizzes based on the user input
- OpenAI GPT

### Database

- Store quiz results for future results
- PostgreSQL

## [High-Level Design]

