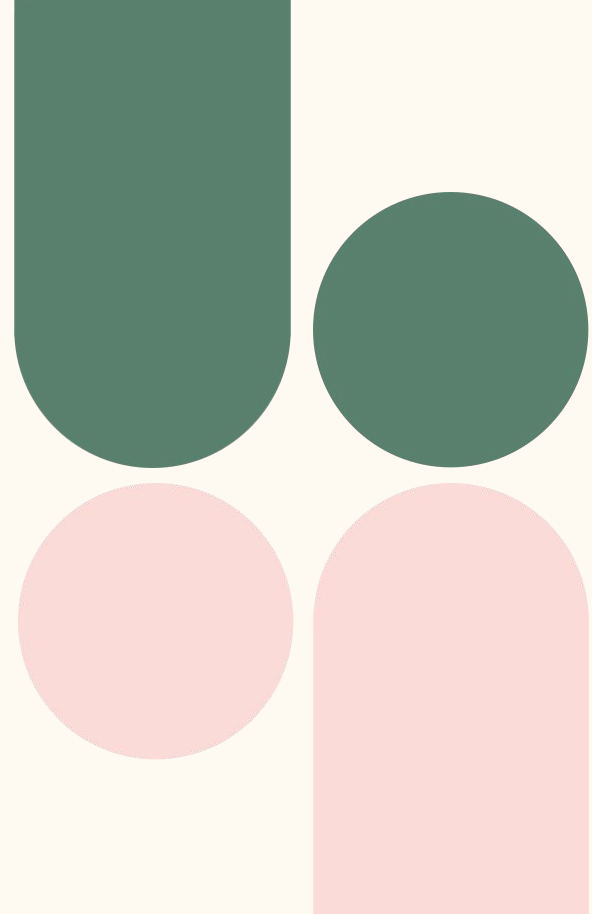


Team 2

CodingCAT

Team Project Final Presentation



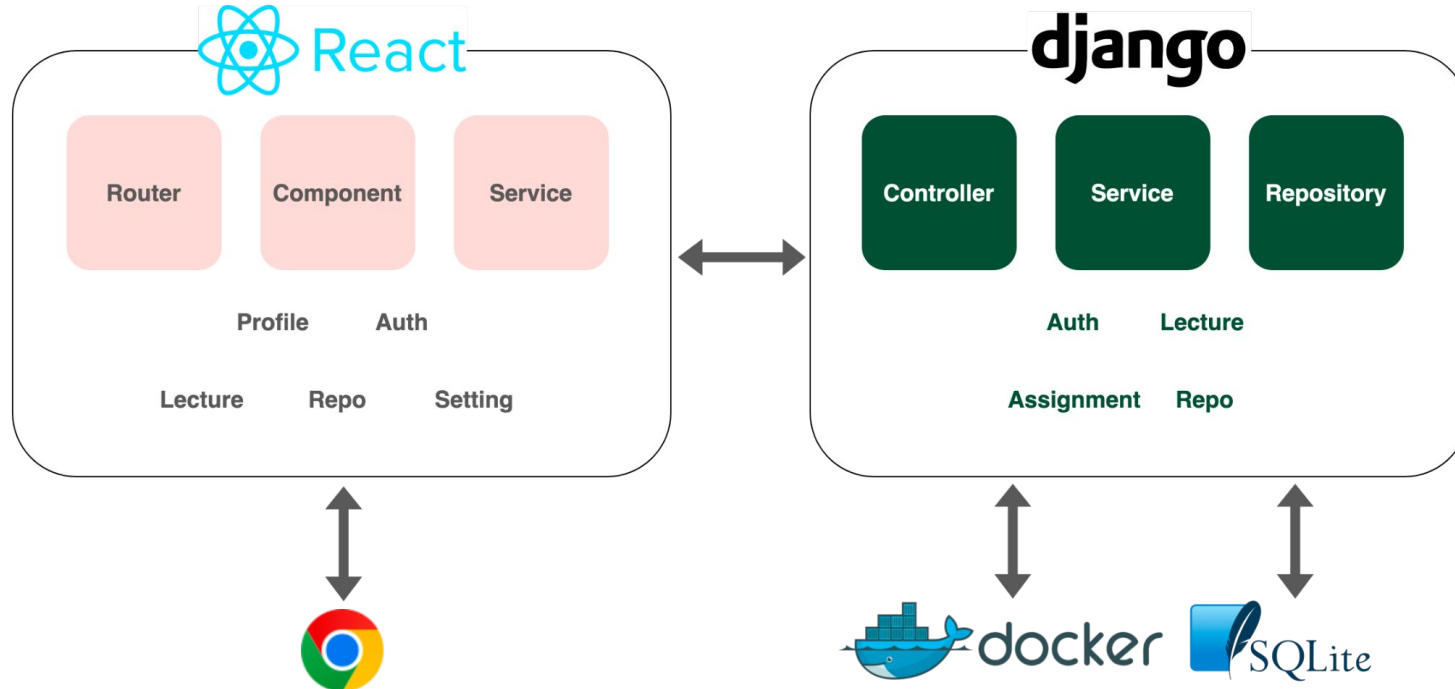
CONTENTS

Team Project Final Presentation

- 1 Overview** Architecture | Interfaces
- 2 Goal** Objectives
- 3 Use Cases** Focused Features
- 4 UI/UX Scenarios** Focused UI/UX
- 5 Deployment** Firebase | AWS
- 6 Team Works** Collaboration | Role

01 Overview

Brief System Architecture

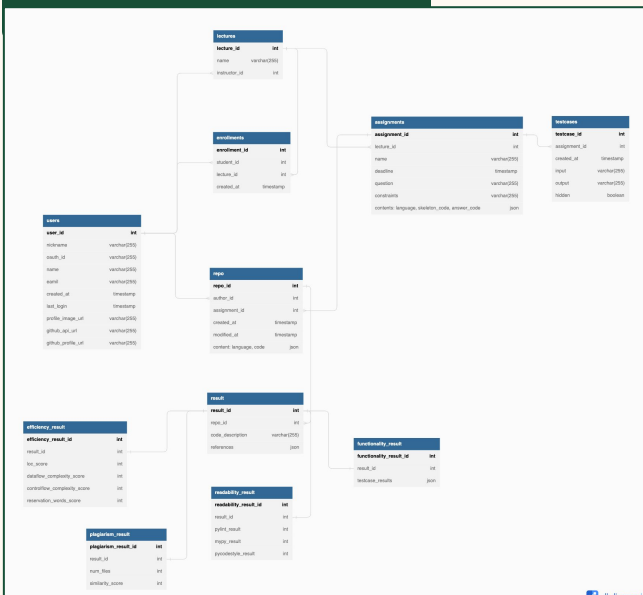


01 Overview

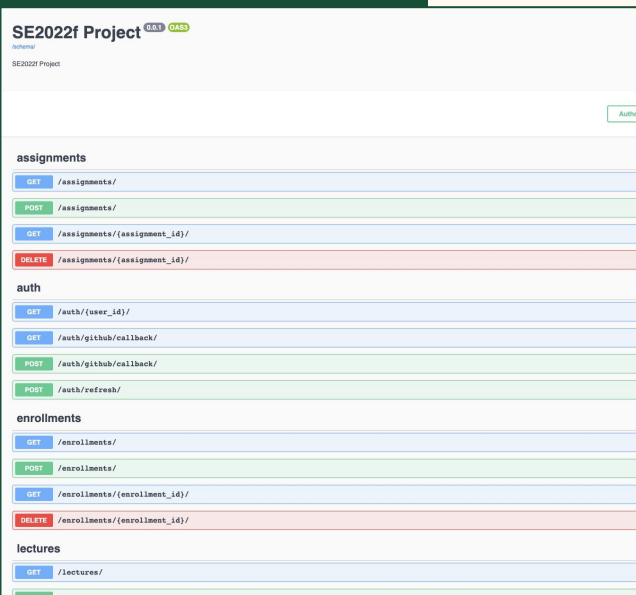
Application Interface Specs.

- First Design Database Entity
- Follow **REST API Architecture** and Document **Autonomously** by Swagger & ReDoc

Entity Schema



Swagger



[Brief API Schema Link](#)

02 Goal

Feature Objectives



Authentication

- Login as Student or Instructor
- By Github OAuth 2.0



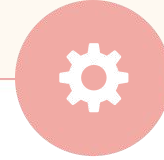
Code Editor & Execution

- Robust Code Editing
- Seamless Executing
- Support Multi-Languages



Test Cases

- Testable to Verify Code
- Compare with Public Cases

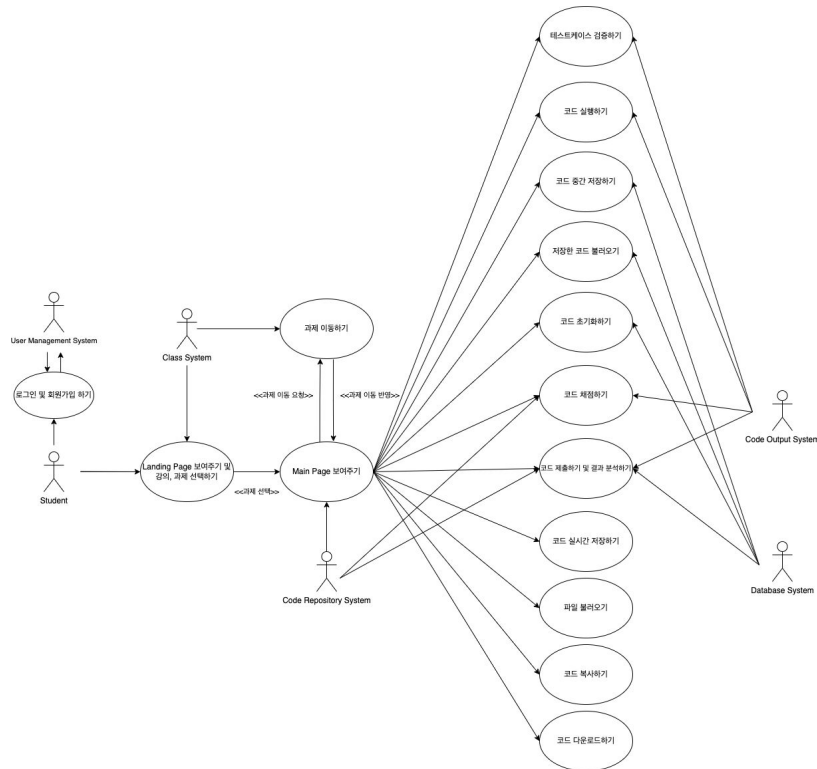


Results

- Show Diverse Scores
- Provide Additional Info

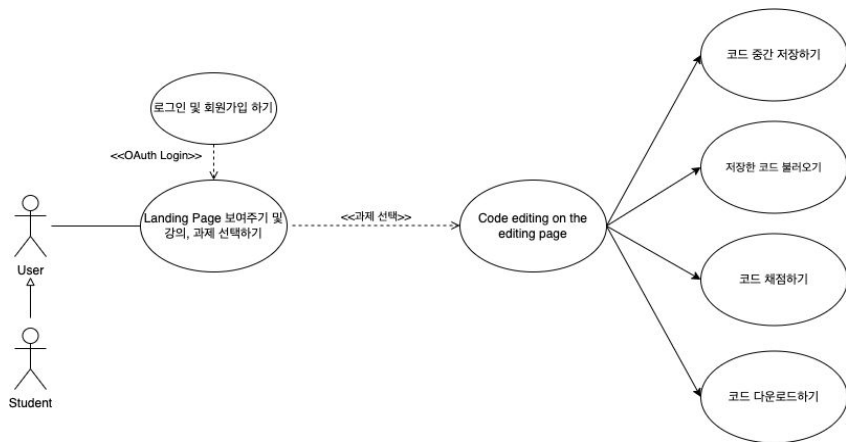
03 Use Cases

Overall Use Case



03 Use Cases

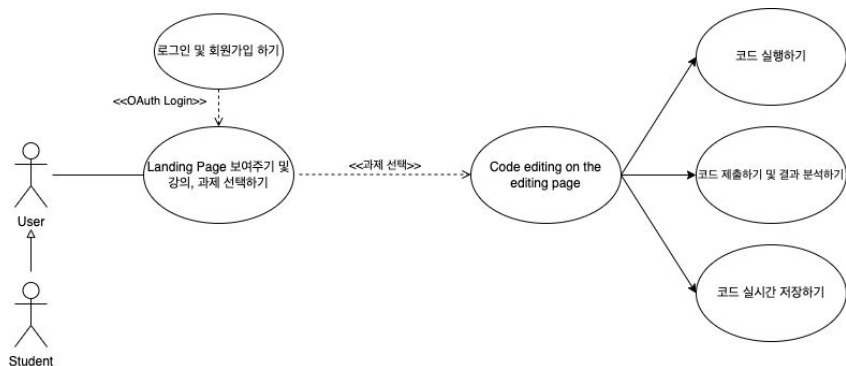
Focused: **Code Editing** and **Save the Work** for a Further Elaboration



- After the OAuth login, the individualized assignment page is provided
- A user picks an assignment and write lines of code
- After the code grading, the user finds there's a room for improvement
- The user determines to elaborate the work later, so save it on the repository
- Also for backing up purpose, the user download the work to its' local space

03 Use Cases

Focused: **Code Editing** and **Submitting an Assignment Directly**

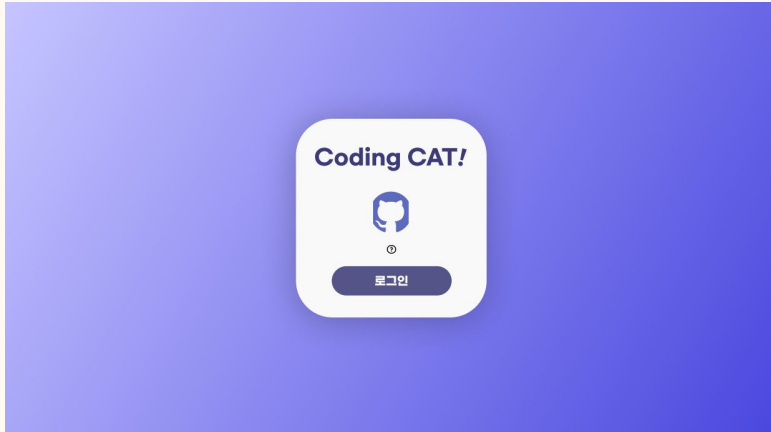


- After the OAuth login, the individualized assignment page is provided
- A user picks an assignment and write lines of code
- **After the code execution, the user finds everything is fine**
- The user determines to submit the assignment, and checks the result
- The code that the user have written is kept through a own workspace

04 UI/UX Scenarios

Functionality Requirements: **Authentication**

Login Page



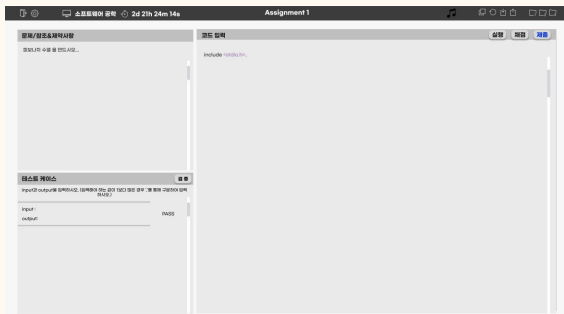
Main Page



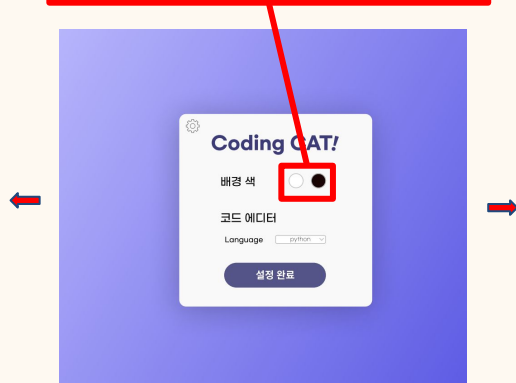
04 UI/UX Scenarios

Functionalities: Light and Dark Mode

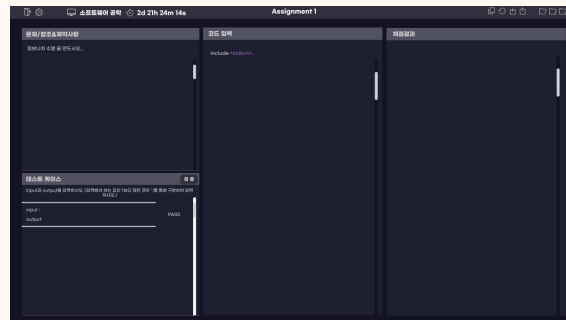
Light mode



change between light or dark mode



Dark mode



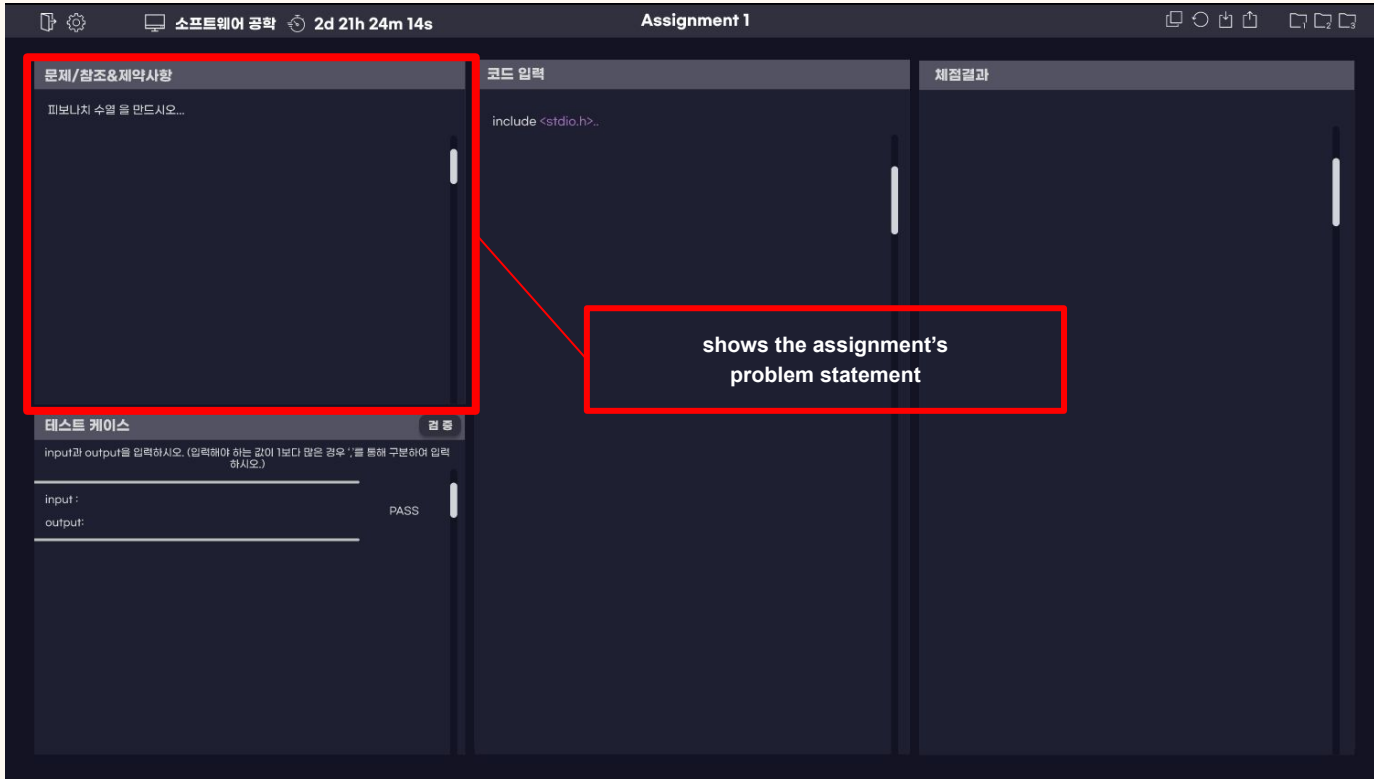
04 UI/UX Scenarios

Functionality Requirements: **Assignments**

문제	강의	공지사항	도움말	게시판	그룹
강의 목록					
소프트웨어 공학 개론	Assignment 1	20h 30m 29s			
	Assignment 2	2d 20h 30m 29s			
프로그래밍 언어	Assignment 1	20h 30m 29s			
	Assignment 2	2d 20h 30m 29s			
	Assignment 3	12d 22h 50m 29s			
프로그래밍 언어	Assignment 1	20h 30m 29s			
	Assignment 2	2d 20h 30m 29s			

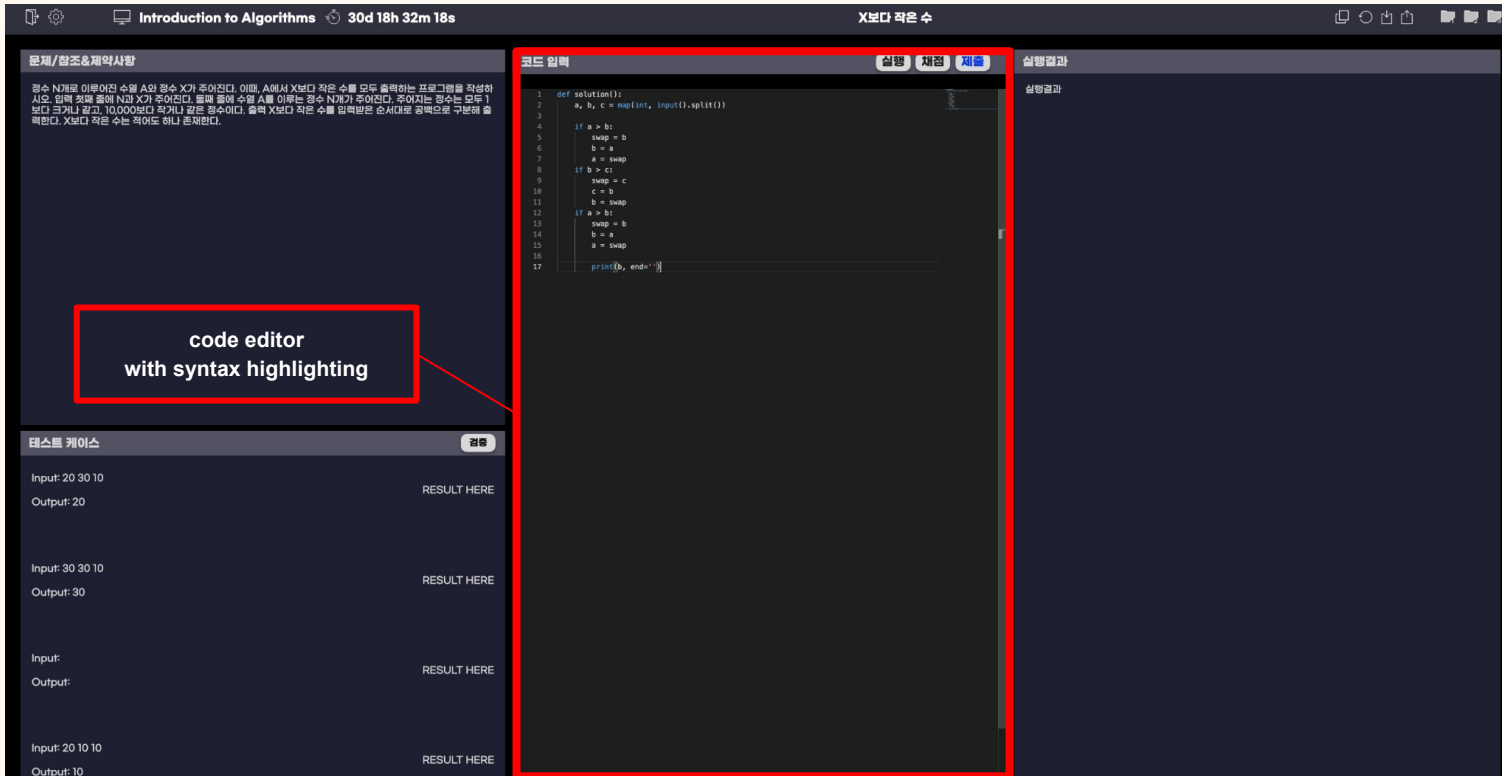
04 UI/UX Scenarios

Functionality Requirements: Problem Display



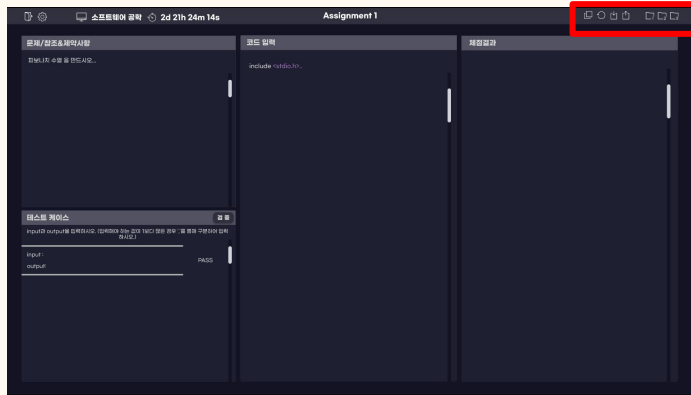
04 UI/UX Scenarios

Functionality Requirements: Code Editing



04 UI/UX Scenarios

Functionality Requirements: Load, Save, and Download



upload code as a source file

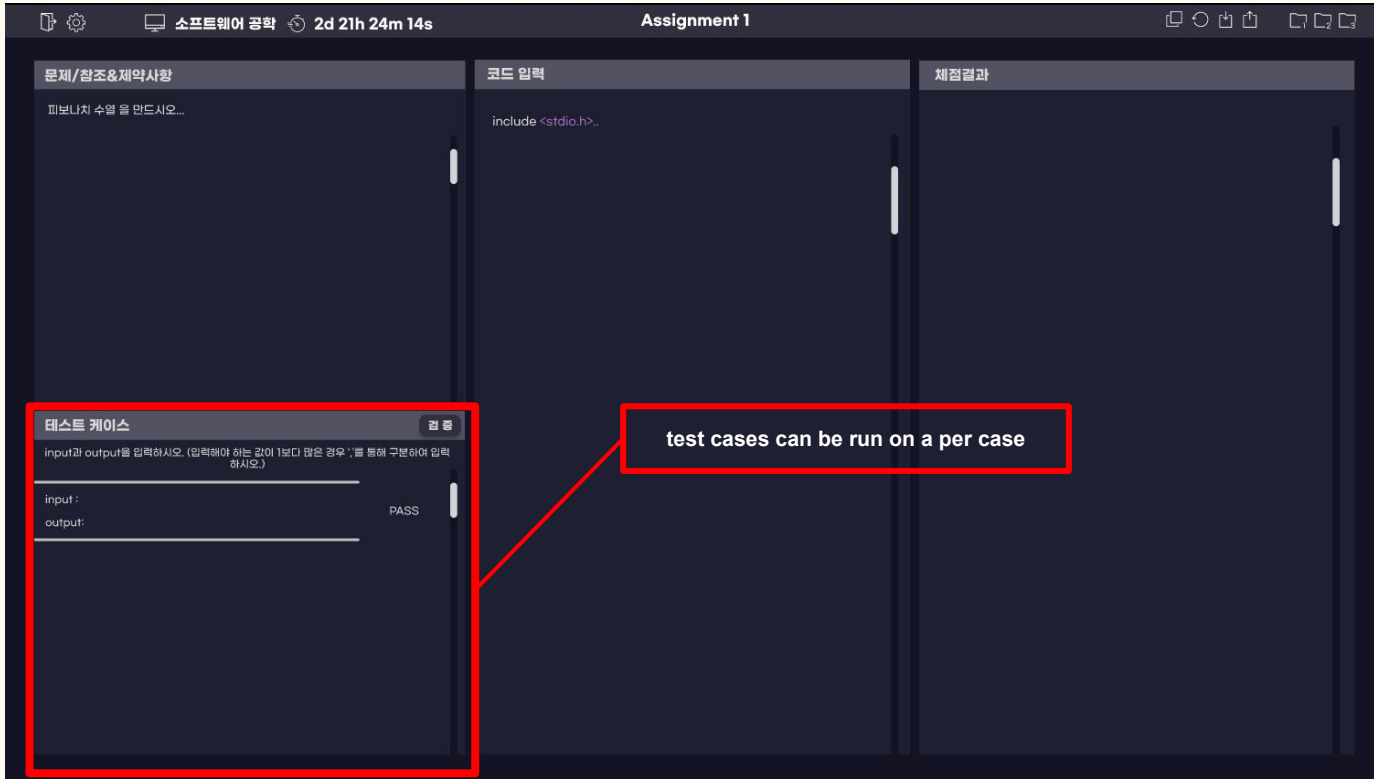


download code as a source file

save current code into the website's repository

04 UI/UX Scenarios

Functionality Requirements: Test Cases and Scoring



04 UI/UX Scenarios

Functionality Requirements: Test Cases and Scoring

The screenshot displays a web application interface for "Assignment 1". The interface is divided into three main panels:

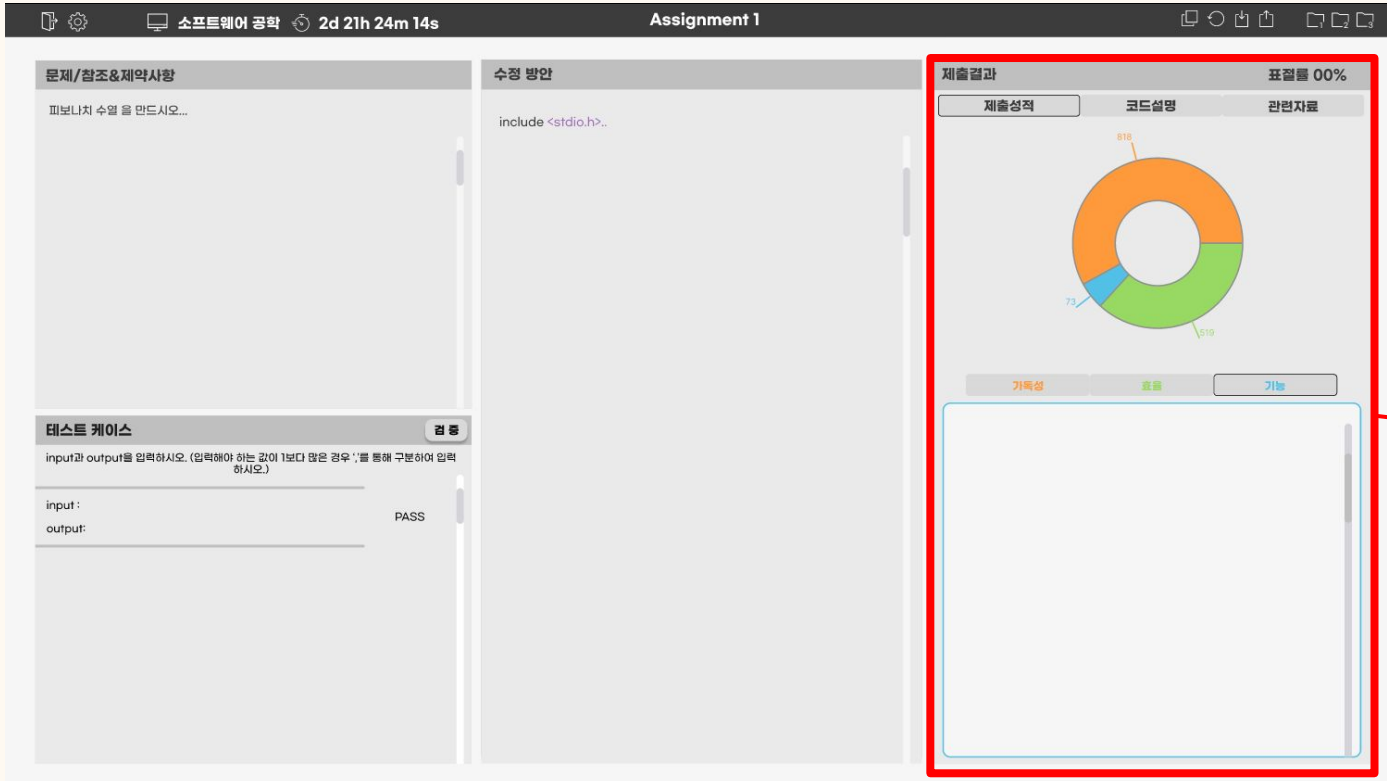
- 문제/참조&제약사항 (Problem/Reference&Constraints):** Contains the problem description: "피보나치 수열 을 만드시오..." (Create a Fibonacci sequence...).
- 코드 확인 (Code Check):** Contains the code snippet: `include <stdio.h>..`
- 채점 결과 (Scoring Results):** This panel is highlighted with a red box. It displays the following information:
 - terminal >> python solution
 - 총점 : 60점
 - 공개 테스트 케이스 1 : Pass
 - 공개 테스트 케이스 2 : Fail
 - 히든 테스트 케이스 3 : Pass
 - 히든 테스트 케이스 4 : Fail
 - 공개 테스트 케이스 5 : Pass

Below the problem description, there is a section for "테스트 케이스 (Test Cases)" with a "검증 (Verify)" button. It shows an input/output example where the input is "input:" and the output is "output:", with a "PASS" status.

A red callout box with the text "show score" points to the "채점 결과" panel.

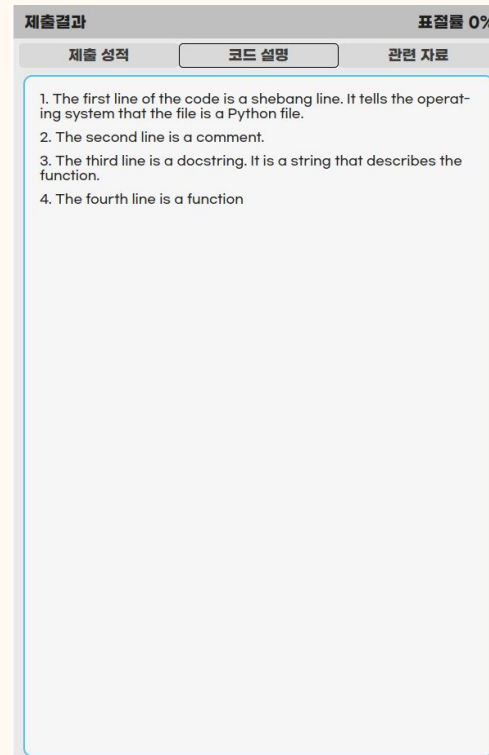
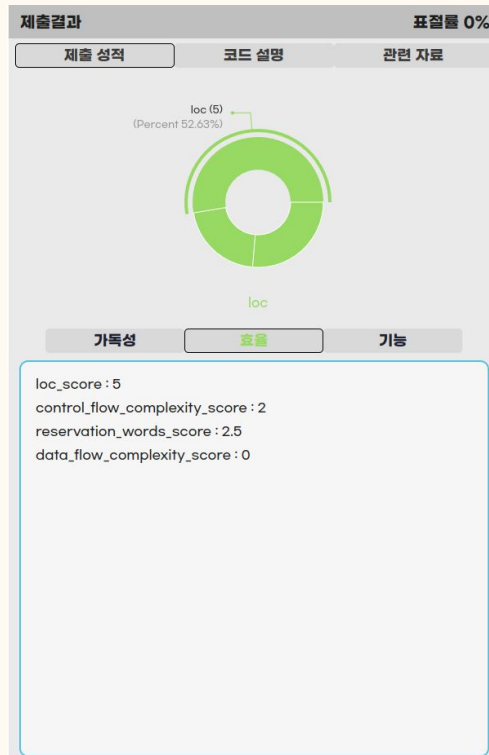
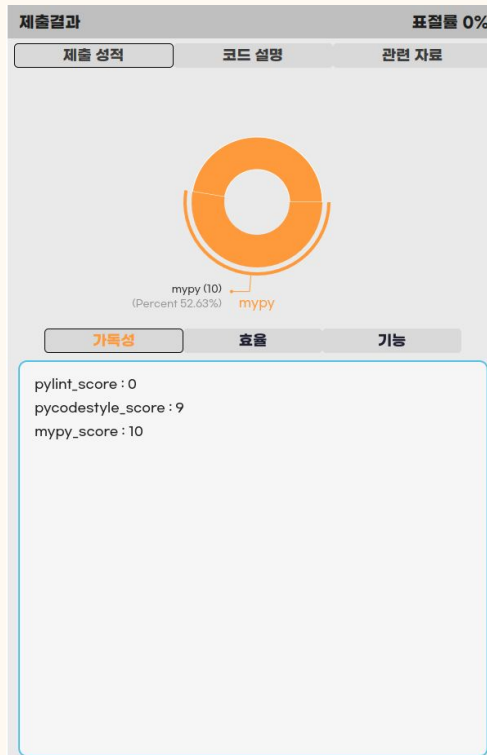
04 UI/UX Scenarios

Functionality Requirements: **Submission Results**



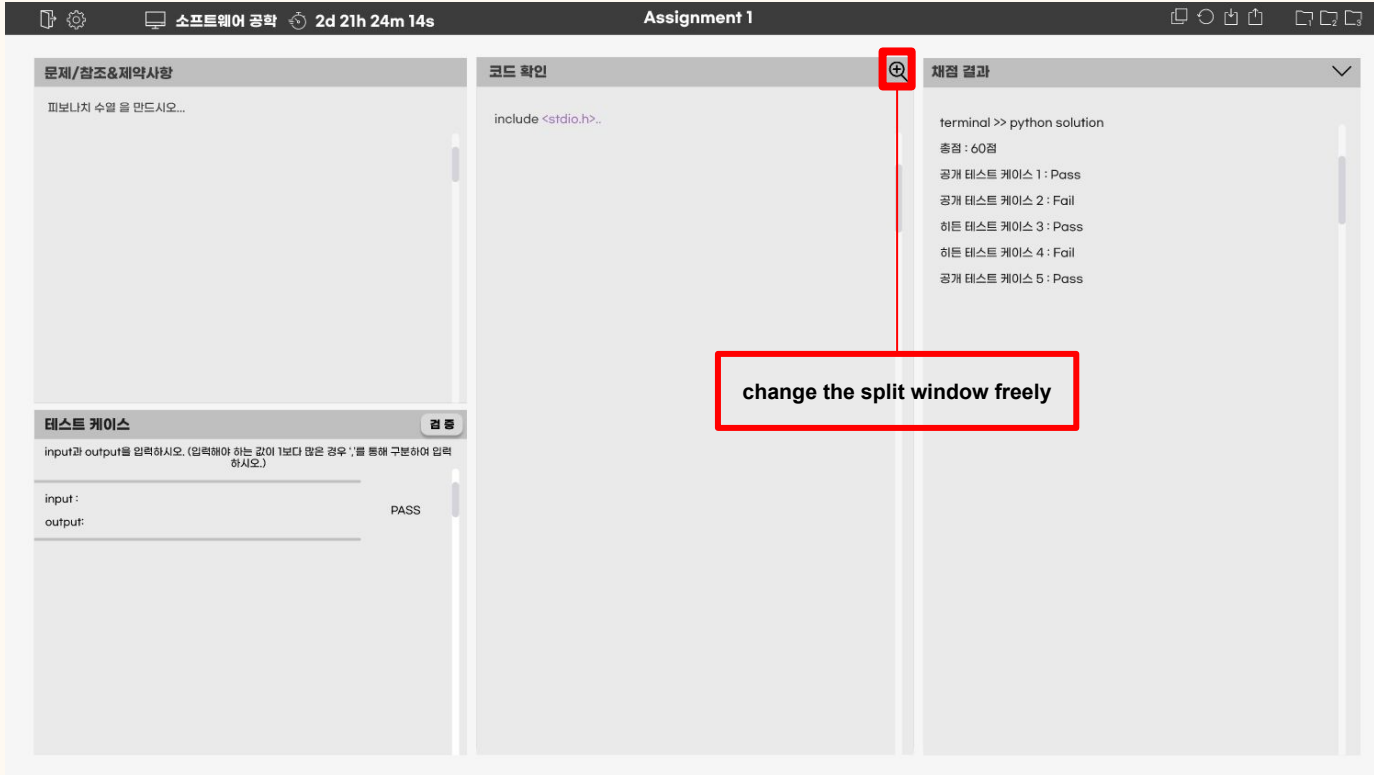
04 UI/UX Scenarios

Functionality Requirements: Submission Results



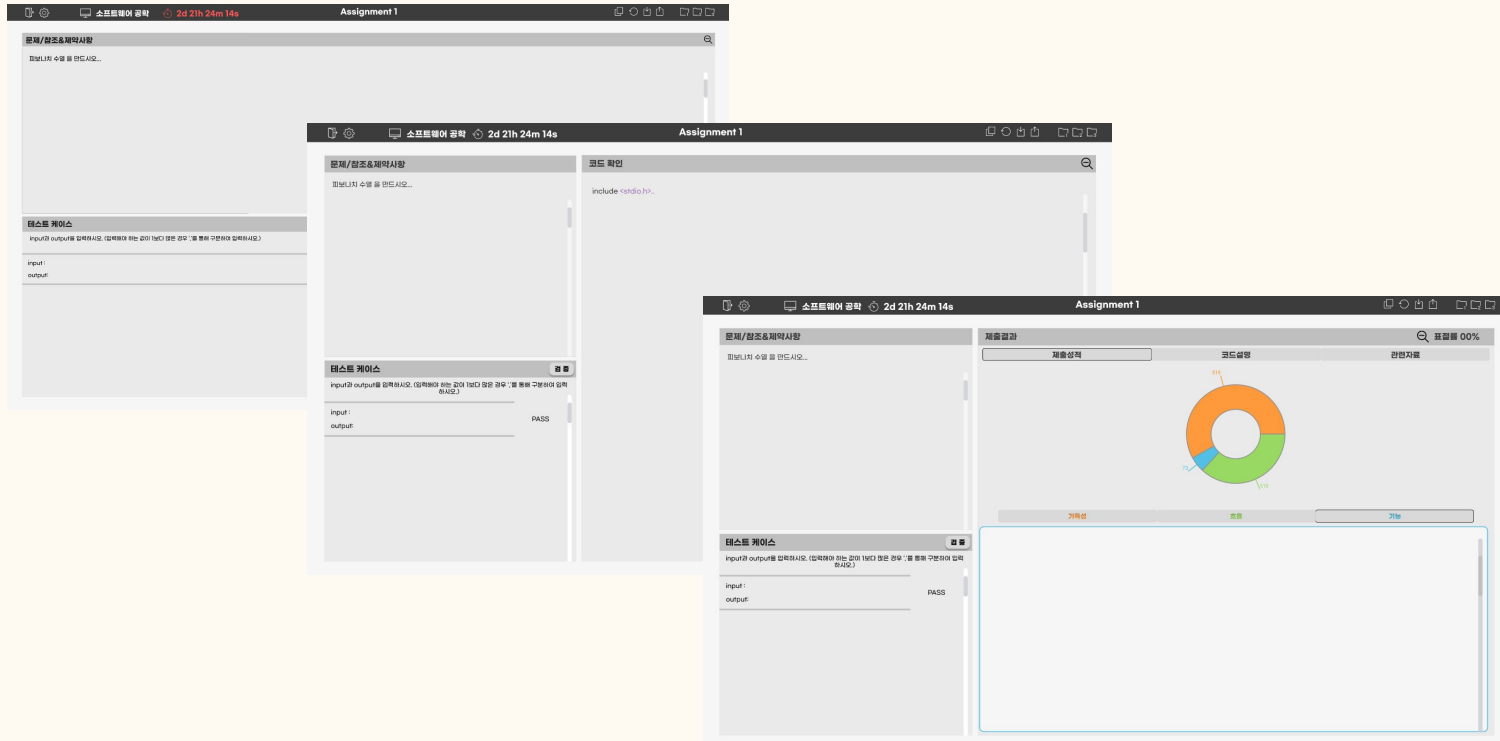
04 UI/UX Scenarios

Extra Functionalities : Page Magnifier



04 UI/UX Scenarios

Extra Functionalities : Page Magnifier



04 UI/UX Scenarios

Extra Functionalities: BGM

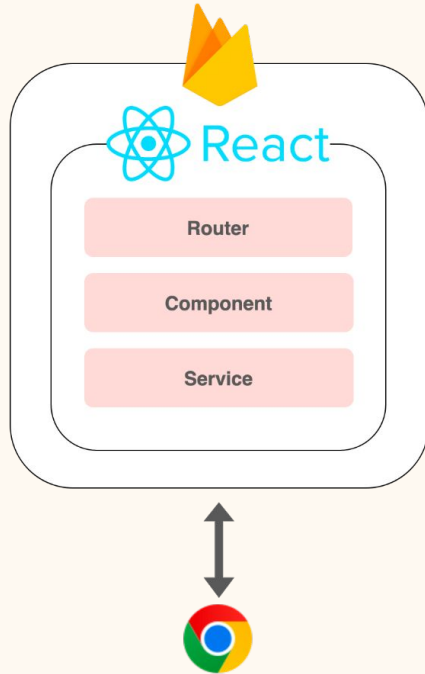
upon clicking the music icon



05 Deployment

Using Firebase & Amazon Web Service

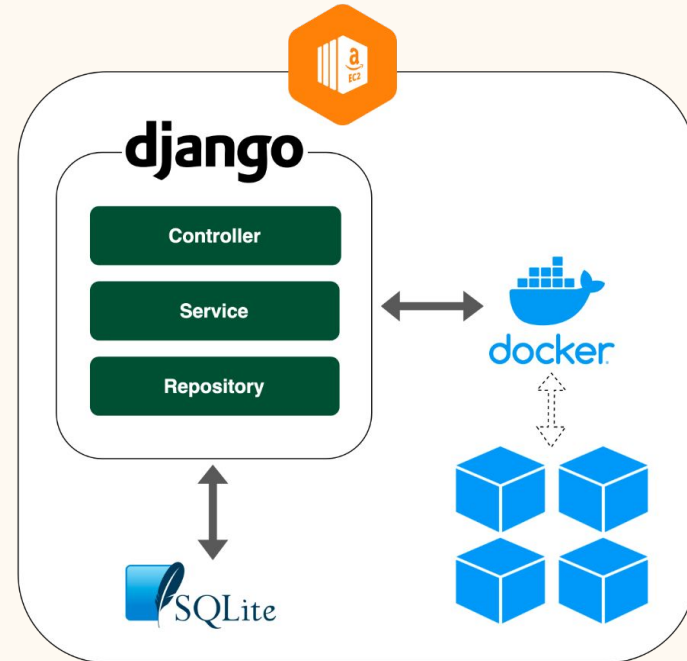
- Deploy Frontend via Firebase



• Accessible Restricted Host IP



- Deploy Backend via AWS



06 Team Statistics

Communication, Collaboration

Period

09.18 ~ 12.02

Slack Messages

1800+

Commits

280+

Branches

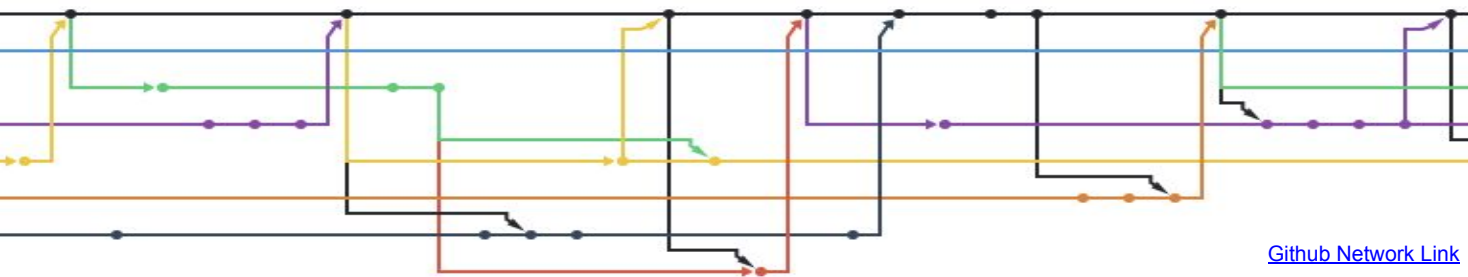
15+

Issues

10+

Pull Requests

50+



[Github Network Link](#)



06 Team Development

Roles and Schedule

Week		4	5	6	7	8	9	10	11	12	13	14	15
Overall Design	UI/UX Design												
	Entity Design												
	API Interface Design												
Requirement Specification													
Design Specification													
Implement & Unit / Integration Testing													
Validation													
Deployment & Release													

UI/UX Design

한태욱

Front-end

신주영

장병우

Deploy

박승호

장병우

Back-end

김동우

박승호

어영준



발표 경청해 주셔서 감사합니다.

Team 2 김동우 신주영 박승호 어영준 장병우 한태욱