

## 캡스톤디자인 II 중간보고서

프로젝트 명 : 초보자를 위한 코딩 스터디 플랫폼

# 캡스톤 디자인 II, 중간보고서

Version 1.0

개발 팀원 명(팀리더) : 김재연  
이주민  
정지윤  
홍유정

대표 연락처 : 010-2446-9917  
e-mail : 20207125@edu.hanbat.ac.kr

## 1. 요구사항 정의서에 명시된 기능에 대하여 현재까지 진척된 결과 및 그 내용을 기술하시오.

### - 소스코드 일부 (코드 작성 지침의 준수 확인용)

#### (1) 코딩 테스트 기능

① 테스트 목록 불러오기: 코딩 테스트 문제 목록을 가져와 표시함.

```
@coding_test.route("/test_list")
@login_required
def test_list():
    page = request.args.get("page", 1, type=int)

    # (중략) 코드 필터링(문제의 언어 & 난이도 선택), 페이지네이션 등
    q_list = conn.query(QList.q_id, QList.q_level, QList.q_name, QList.q_lang)\
        .filter(and_(*filters))\
        .offset((page - 1) * PER_PAGE)\
        .limit(PER_PAGE)\
        .all()

    conn.close()

    return render_template(
        "test_list.html",
        q_list=q_list,
        current_page=page,
        total_pages=(total_tests + PER_PAGE - 1) // PER_PAGE
    )
```

② 코드 컴파일과 실행: 사용자가 작성한 코드를 실행(컴파일) 하고 결과를 반환함.

```
@coding_test.route("/compile", methods=["POST"])
@csrf.exempt
def compile():
    code = request.form.get("code")
    language = request.form.get("language")

    session["language"] = language # 여기에서 언어 정보를 세션에 저장

    if language == "python":
        output_str = python_run_code(code)
    elif language == "c":
        output_str = c_compile_code(code)
    elif language == "c++":
        output_str = cpp_compile_code(code)
    elif language == "java":
        output_str = java_compile_run_code(code)

    return output_str
```

③ 코드 저장: 작성한 코드와 컴파일 결과, 채점 결과 등을 데이터베이스에 저장함.

```

@coding_test.route("/save_code", methods=["POST"])
@csrf.exempt
def code_save():
    try:
        db_session = get_db_connection()

        q_id = request.form.get("q_id")
        # (중략) HTTP POST 요청에서 폼 데이터를 가져옴
        # 폼 데이터: 사용자 id, 코드 내용, 사용한 언어, 컴파일 결과, 정답 여부
        # 데이터베이스에 저장
        new_submission = CodeSubmission(
            q_id=q_id,
            user_id=user_id,
            code_content=code_content,
            language=language,
            compile_result=compile_result,
            is_correct=is_correct,
        )

        db_session.add(new_submission)
        db_session.commit()
        db_session.close()

        return jsonify({"message": "Code saved successfully!"})

    except Exception as e:
        return jsonify({"error": str(e)})

```

## (2) 로그인/회원가입 기능

① 로그인: 사용자 정보를 검증하고 세션을 생성함.

```

@auth.route("/login", methods=["GET", "POST"])
def login():
    # 로그인 상태를 확인하여 이미 로그인한 경우 홈으로 리다이렉트
    if current_user.is_authenticated:
        return redirect(url_for("home"))

    # ... (중략: 폼 객체 생성, 에러 메시지 초기화)

    # POST 요청 시 폼 유효성 검사
    if request.method == "POST" and form.validate_on_submit():
        db_session = get_db_connection()

        # 사용자가 입력한 이메일과 비밀번호를 가져옴
        email = form.useremail.data
        password = form.password.data

        # 해당 이메일을 가진 사용자를 데이터베이스에서 찾음
        user = db_session.query(User).filter_by(user_email=email).first()

        # 사용자가 존재하고 비밀번호가 맞는 경우
        if user is not None:
            password_check = user.check_password(password)

```

```

        login_user(user) # 사용자가 로그인
        return redirect(url_for("home"))

# ... (에러 메시지 설정 등은 중략)

```

② 회원가입: 사용자 정보를 데이터베이스에 저장함.

```

@auth.route("/signup", methods=["GET", "POST"])
def signup():
    if current_user.is_authenticated:
        return redirect(url_for("home"))

    # ... (폼 객체 생성과 CSRF 토큰 설정 등은 중략)

    # 폼 유효성 검사를 통과한 경우
    if form.validate_on_submit():

        # 사용자가 입력한 이름, 이메일, 비밀번호를 가져옴
        name = form.name.data
        email = form.useremail.data
        password = form.password.data

        # 새로운 user 생성
        new_user = User(username=name, user_email=email)
        new_user.set_password(password)

        # db에 유저 저장
        db_session.add(new_user)
        db_session.commit()

    # ... (세션 클로즈 및 로그인 페이지로 리다이렉트 등은 중략)

```

(3) 게시판 기능

① 게시글 작성

```

@board.route("/board_write", methods=["GET", "POST"])
def board_write():
    form = BoardWriteForm() # 폼 객체 생성

    # 폼 유효성 검사
    if form.validate_on_submit():
        title = form.title.data
        content = form.content.data

        # 파일 업로드 처리 부분
        for file in files:
            if file and file.filename:
                original_filename = file.filename
                safe_filename = unique_filename(original_filename)
                file.save(os.path.join(UPLOAD_PATH, safe_filename))
                file_paths.append(safe_filename)

```

```

# DB에 게시물 추가
new_post = Board(
    user_id=current_user.id, title=title, content=content, file_path=file_paths, view=0
)

db_session = get_db_connection()
try:
    db_session.add(new_post)
    db_session.commit()

# ...(중략: 에러 처리 및 리다이렉션 부분)

```

## ② 게시물 수정

```

@board.route("/board_edit/<int:board_id>", methods=["GET", "POST"])
def board_edit(board_id):
    form = BoardEditForm()

    db_session = get_db_connection()
    board_instance = db_session.query(Board).filter(Board.board_id == board_id).first()

    # 폼 유효성 검사
    if form.validate_on_submit():
        board_instance.title = form.title.data
        board_instance.content = form.content.data

    # ...(중략: 파일 업로드 및 삭제 처리)

    try:
        db_session.commit() # DB에 변경 사항 저장

    # ...(중략: 에러 처리 및 리다이렉션)

```

## ③ 게시물 삭제

```

@board.route("/board_delete/<int:board_id>")
def board_delete(board_id):
    db_session = get_db_connection()
    board_instance = db_session.query(Board).filter(Board.board_id == board_id).first()

    if board_instance.user_id != current_user.id:
        return redirect(url_for("board.board_detail", board_id=board_id, message="삭제 권한이 없습니다."))

    try:
        db_session.delete(board_instance) # 게시물 삭제
        db_session.commit() # db 변경 사항 저장

    # ... (중략: 에러 처리 및 redirect 로직)

```

## (4) 코딩 게임 기능

- ① 게임 데이터 가져오기: 게임에 필요한 데이터를 DB로부터 무작위 또는 전체로 가져옴.
  - Typing Game 문제 가져오기



```

@game.route("/api/get_typinggame_questions")
@csrf.exempt
def get_typinggame_questions():
    conn = get_db_connection()
    # 랜덤으로 문제 선택
    typinggame_questions = conn.query(TypingGame).order_by(func.random()).first()
    conn.close()

    return jsonify(
        {
            "code": typinggame_questions.code,
            "language": typinggame_questions.language,
            "description": typinggame_questions.description,
        }
    )

```

- Drag game 문제 가져오기

```

@game.route("/api/get_draggame_questions")
@csrf.exempt
def get_draggame_questions():
    conn = get_db_connection()

    # 모든 draggame 문제를 가져옴
    draggame_questions = conn.query(DragGame).all()

    # ... (중략: 문제들을 JSON 형태로 변환)

    return jsonify(allQuestions)

```

② 게임 점수 저장: 게임을 마친 후에 사용자의 점수를 DB에 저장함.

```

@game_score.route("/api/save_game_result", methods=["POST"])
@csrf.exempt
def save_game_result():
    data = request.json

    score = data["score"]
    language = data["language"]
    gameType = data["gameType"]

    conn = get_db_connection()

    if gameType == "drag":
        game_score = DragGameScore(user_id=current_user.id, drag_language=language,
                                   drag_score=score, played_at=current_kst_time,
                                   )
    elif gameType == "output":
        game_score = OutputGameScore(user_id=current_user.id, output_language=language,
                                     output_score=score, played_at=current_kst_time,
                                     )

```

```

else:
    return jsonify({"message": "Invalid game type!"}), 400

conn.add(game_score)
conn.commit()

return jsonify({"message": "Game result saved successfully!"})

```

③ 리더보드: 게임 타입과 언어에 따라 최고 점수를 얻은 사용자를 목록으로 제공함.

```

@game_score.route("/get_leaderboard", methods=["GET"])
def get_leaderboard():
    conn = get_db_connection()
    game_type = request.args.get("game_type")

    # ... (중략: 게임 타입 및 언어 설정)
    leaderboard_dict = {}

    for lang in languages:
        if game_type == "outputgame":
            leaderboard_data = (
                conn.query(
                    User.username,
                    OutputGameScore.output_language,
                    OutputGameScore.output_score,
                    OutputGameScore.played_at,
                )
                .filter(OutputGameScore.output_language == lang)
                .join(User, User.id == OutputGameScore.user_id)
                .order_by(OutputGameScore.output_score.desc(), OutputGameScore.played_at.asc())
                .limit(5)
                .all()
            )
            leaderboard_key = lang
        else: # ... (중략: draggame)
            leaderboard_key = lang

        leaderboard_dict[leaderboard_key] = [
            {
                "username": item.username,
                "language": item.drag_language if game_type == "draggame" else item.output_language,
                "score": item.drag_score if game_type == "draggame" else item.output_score,
                "played_at": item.played_at.strftime("%Y-%m-%d %H:%M:%S"),
            }
            for item in leaderboard_data
        ]

    return jsonify(leaderboard_dict)

```

(5) 마이페이지 기능

① 마이페이지: 로그인한 사용자의 게시물, 작성한 코드, 댓글, 게임 기록을 가져와 보여줌.

```

@user_profile.route("/mypage")
@login_required
def mypage():
    db_session = get_db_connection()

```

```

# 사용자의 게시물 가져오기
user_posts = db_session.query(Board).filter_by(user_id=current_user.id).all()

# 사용자가 작성한 코드 정보 가져오기
user_codes = (
    db_session.query(CodeSubmission, QList)
    .join(QList, CodeSubmission.q_id == QList.q_id)
    .filter(CodeSubmission.user_id == current_user.id)
    .all()
)

# 사용자가 작성한 댓글 정보 가져오기
user_comments = db_session.query(Comments).filter_by(user_id=current_user.id).all()

# 게임 기록 가져오기
output_game_scores = db_session.query(OutputGameScore).filter_by(user_id=current_user.id).all()
drag_game_scores = db_session.query(DragGameScore).filter_by(user_id=current_user.id).all()

# ... (중략: 게임 결과 처리 )
db_session.close()

return render_template(
    "mypage.html",
    user_posts=user_posts,
    user_codes=user_codes,
    user_comments=user_comments,
    game_results=game_results, # 추가된 게임 결과 리스트
)

```

## (6) 피드백 관련 기능

- ① GPT API를 통한 코드 피드백: 사용자가 작성한 코드를 GPT API를 통해 피드백을 받음.

```

@ai_feedback.route("/feedback")
@csrf.exempt
def feedback():
    code = request.form.get("code")
    language = request.form.get("language")

    conn = get_db_connection()
    q_info = conn.query(QList).filter(QList.q_id == session["q_id"]).first()

    # GPT API를 호출하여 피드백 받기
    feedback = get_feedback(problem_description, code, language)

    conn.close()
    return feedback

```

- ② AI 챗봇: 사용자가 실시간으로 질문할 수 있는 페이지를 제공하고 응답을 받음.

```

@ai_feedback.route("/ai_chatbot_submit", methods=["GET", "POST"])
@csrf.exempt
def ai_chatbot_submit():
    if request.method == "POST":

```



```

content = request.form["userText"]

# GPT API를 호출하여 응답 받기
chatbot_response = generate_response(content)

return chatbot_response

```

## - ver1.0 실행파일(환경 설치 문서 포함)

### (1) 개요

- 프로젝트를 실행하기 위한 환경 설정에 관해 설명한다. 프로젝트는 Docker를 사용하여 로컬 및 서버 환경에서 실행할 수 있으며, AWS와 RDS(MySQL)을 사용하여 웹 서버를 호스팅 한다.
- 전제 조건: Docker가 이미 설치되어 있음을 전제로 한다.

### (2) Docker 설정

#### ① Docker

- Base Image: python:3.9-slim-buster
- 작업 디렉터리: /app
- 패키지 설치: default-libmysqlclient-dev, g++, openjdk-11-jdk
- Python 라이브러리 설치: requirements.txt을 통한 설치

#### ② docker-compose.yml

- Version: '3.8'
- Services: app
- Ports: 5000 (host) : 5000 (container)
- Environment Variables: DB\_HOST

### (3) 실행 방법

#### ① 로컬 환경

1. Git을 사용해 프로젝트를 클론 한다.  
→ `git clone https://github.com/HBNU-SWUNIV/come-capstone23-pse.git`
2. 프로젝트 폴더로 이동한 뒤, 서비스를 실행한다.  
→ `cd "come-capstone23-pse\003 Code\codepse"`  
`docker-compose up`
3. 브라우저에서 `http://localhost:5000`으로 접속하여 애플리케이션을 확인한다.

#### ② AWS 환경

1. AWS EC2 인스턴스를 생성하고 SSH로 접속한다.
2. Git을 사용해 프로젝트를 클론 한다.

```
→ git clone https://github.com/HBNU-SWUNIV/come-capstone23-pse.git
3. 프로젝트 폴더로 이동한 뒤, 서비스를 실행한다.
→ cd "come-capstone23-pse\003 Code\codepse"
   docker-compose up
```

(4) AWS RDS(MySQL) 설정

- AWS RDS 콘솔에서 MySQL 데이터베이스 인스턴스를 생성한다.
- 보안 그룹과 VPC를 설정하여 EC2와 통신을 할 수 있게 한다.
- 생성된 RDS 인스턴스의 엔드 포인트를 docker-compose.yml의 DB\_HOST 환경 변수에 설정한다.

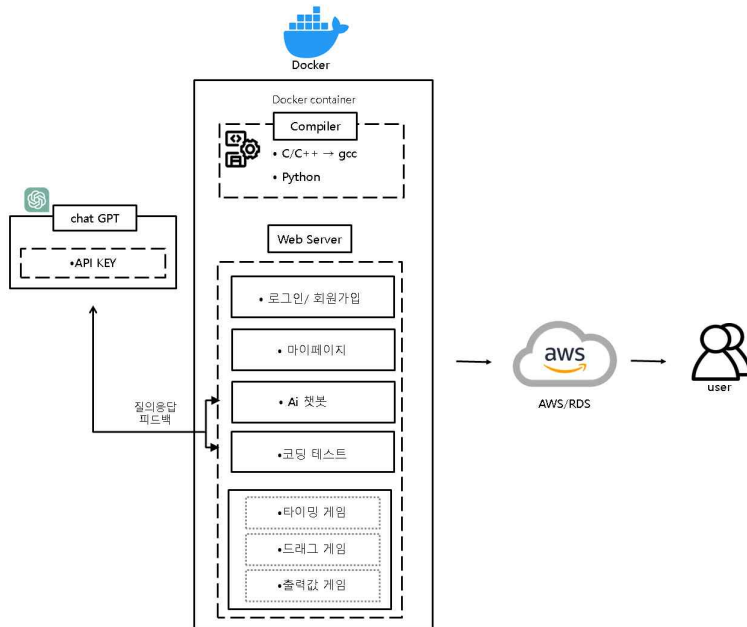
## - 시스템 구성도 :

클라이언트에서 도메인 요청 시 응답

IP 주소에 HTML 요청 시 응답

AWS의 RDS를 사용한 데이터베이스 구축 완료

AWS를 사용하여 서버 구축



## - 유스 케이스 :

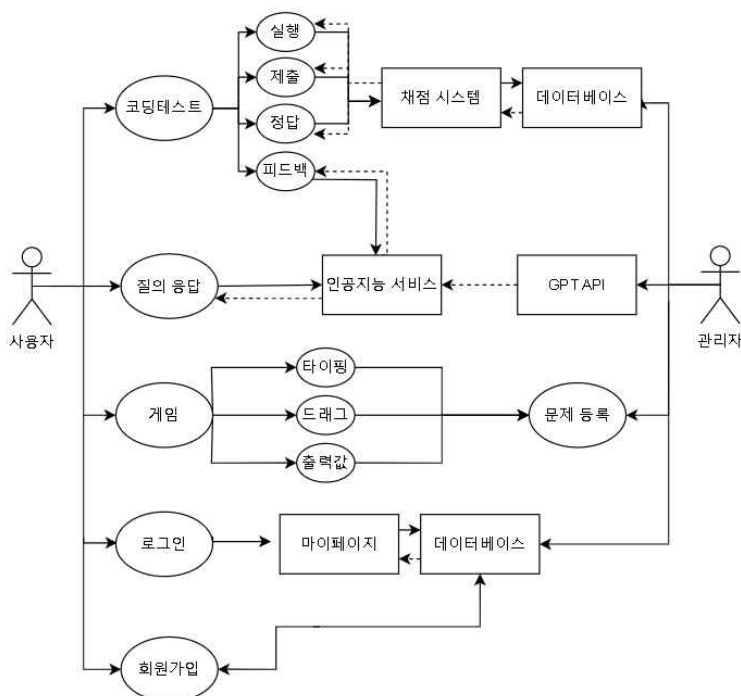
회원가입 및 로그인 기능 RDS와 연결하여 구현

코딩 스터디를 위한 채점 서버 구축, 데이터베이스에 문제 저장 후 불러오는 기능 구현

GPT API를 이용한 인공지능 질의응답 기능 구현

코드 리뷰 및 커뮤니티 공간 기능 구현

코딩 게임에 대한 문제 데이터베이스 저장 후 불러오는 기능 구현



## 2. 프로젝트 수행을 위해 적용된 추진전략, 수행 방법의 결과를 작성하고, 만일 적용 과정에서 문제점이 도출되었다면 그 문제를 분석하고 해결 방안을 기술하시오.

### - 팀원의 책임 및 역할 수행에 대한 결과, (문제점, 해결 방안)

자료조사를 통해 프로젝트의 각 파트를 분석하고 업무를 할당하였지만, 우리가 예상하지 못한 어려움이 각 파트 간에 큰 격차로 나타났다. 이러한 문제를 해결하기 위해 주차 별로 프로젝트 진행 상황을 서로 공유하고, 지식과 경험을 공유하며 피드백을 주고받는 방식으로 노력하였다.

프로젝트의 초기에는 파트 간 협력과 공유가 진행 상황을 원활히 유지하는 데 도움이 되었다. 그러나 프로젝트가 진행됨에 따라 각 파트 내에서 더 깊은 이해와 전문성이 필요한 상황이 발생하였다. 이 문제에 대해서 꾸준한 논의를 통해 해결책을 모색하였다.

팀원들끼리 지속적으로 의사소통하고 협력하여, 파트 간의 지식과 역량을 공유하며 프로젝트를 성공적으로 완료하고자 노력하였다. 이런 협력과 논의를 통해 우리 프로젝트는 어려움을 극복하고 더 나은 결과를 얻을 수 있었다.

### - 프로젝트 일정 계획에 맞추지 못한 경우의 문제점, 해결 방안

프로젝트를 진행하면서 각 파트별로 팀원들을 배치하고, 프로젝트의 복잡도를 낮추기 위해 주기적으로 각자의 진행 상황과 지식을 공유하려 노력했다. 그러나 이런 방식은 팀원 간의 상호 피드백 시간이 많이 소요되어 프로젝트 진행에 시간적 압박을 느꼈다.

이 문제를 해결하기 위해, 캡스톤 디자인2의 일정이 초기 2달은 개발에 초점을 맞추고, 그 이후에는 테스트와 유지 보수에 중점을 두는 것을 고려하여, 초기 2달 동안은 상세한 진행 상황 공유와 지식 공유 대신, 팀원 간의 피드백에 더욱 집중하기로 했다.

### - 요구사항 변경 관리의 결과, (문제점, 해결 방안) 등등

- 문제점 : 여러 언어로 동일한 문제를 풀 때, 언어마다의 특성과 문법을 제대로 활용할 수 없었다. 사용자들은 각 언어의 고유한 특징을 반영한 문제를 원했으나 현재의 문제는 그런 부분을 충족시키지 못하였다.

- 해결 방안 : 코딩 테스트의 문제 제공 방식을 수정하여, 각 프로그래밍 언어에 맞는 고유한 문제들을 따로 제공하게 구성하였다. 이를 통해 사용자들이 해당 언어의 특성을 최대한 활용하여 문제를 해결할 수 있을 것이다.

- 기대효과 : 사용자의 요구를 충족시키면서 각 언어의 특성을 잘 반영한 문제를 통해 사용자의 실력을 보다 정확하게 평가할 수 있게 될 것이다.



[별첨1]

프로젝트명 : 초보자를 위한 코딩 스터디 플랫폼

# 소프트웨어 요구사항 정의서

Version 1.0

개발 팀원 명(팀리더) : 김재연

이주민

정지윤

홍유정

대표 연락처 : 010-2446-9917

e-mail : 20207125@edu.hanbat.ac.kr

# 목차

1. 개요 .....	3
2. 시스템 장비 구성 요구사항 .....	4
3. 기능 요구사항 .....	5
4. 성능 요구사항 .....	8
5. 인터페이스 요구사항 .....	9
6. 데이터 요구사항 .....	10
7. 테스트 요구사항 .....	12
8. 보안 요구사항 .....	13
9. 품질 요구사항 .....	15
10. 제약 사항 .....	16
11. 프로젝트 관리 요구사항 .....	17

## 1. 시스템 개요

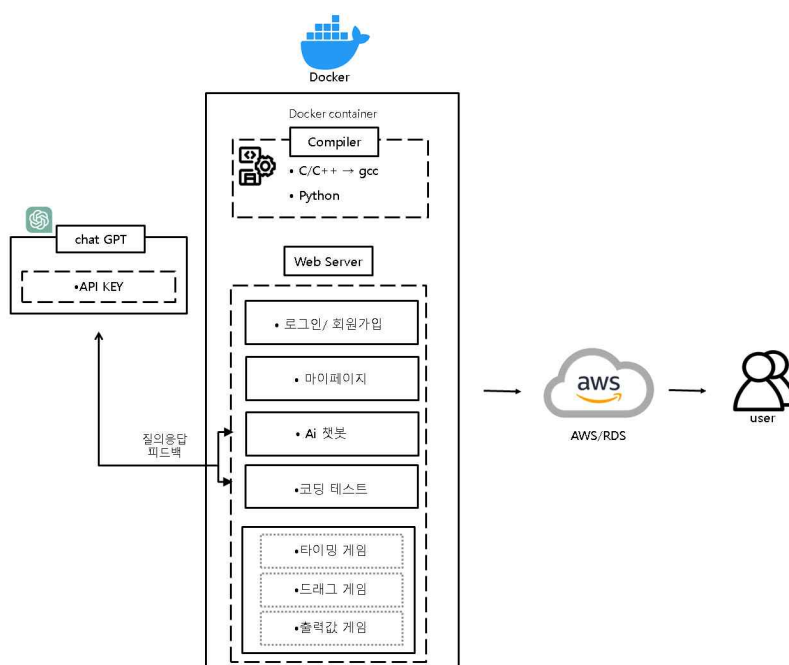
최근 IT의 중요성이 점점 확대되며 프로그래밍 능력이 필수적인 기술로 여겨지고 있다. 그러나 많은 사람들이 프로그래밍 학습의 난이도와 지루함 때문에 흥미를 잃거나 학습을 포기하는 경향이 있다. 이러한 문제점을 해결하기 위해 초보자들을 위한 코딩 스터디 플랫폼을 개발하고자 한다.

이 플랫폼의 핵심 목표는 사용자에게 코딩테스트 준비 및 코딩 학습 환경을 제공하여, 학습 과정을 흥미롭고 효과적으로 만드는 것이다. 플랫폼의 기능은 다음과 같다.

- 회원가입 & 로그인 : 사용자가 서비스에 가입하고 자신의 개인 정보를 등록할 수 있는 기능이다. 로그인 기능을 통해 개인화된 학습 경험을 제공한다.
- 코딩 테스트 & 채점 : 다양한 난이도의 코딩 테스트를 제공하여, 사용자들이 문제 해결 능력을 키울 수 있으며, 채점 서버를 통해 자동으로 채점된다.
- 게임 : 코딩 기초 역량을 쌓기 위한 게임을 제공한다. 이 게임들은 학습 과정을 더욱 즐겁게 만들어줄 것이다.
- 개인화 학습 경험 : 사용자들의 학습 이력을 기록하여, 개인화된 학습 경험을 제공한다. 이를 통해 사용자는 자신의 학습 과정을 더욱 효과적으로 파악할 수 있다.
- 코드 리뷰 : 사용자들이 문제 해결 능력과 코드 작성 기술을 향상하기 위한 기능을 제공한다.
- Chat GPT API지원 : Chat GPT API를 통해 사용자들이 코딩 중에 생기는 의문점을 실시간으로 해결할 수 있다.
- 보안 : 사용자의 개인 정보는 암호화 기술을 통해 안전하게 보호된다.

이러한 기능들은 사용자가 코딩을 학습하는 과정을 더욱 즐겁고 효과적으로 만들어 프로그래밍 능력을 향상할 수 있도록 도움을 준다. 이 플랫폼을 통해 사용자가 코딩을 즐길 수 있고, 필수적인 프로그래밍 기술을 획득할 수 있게 돕고자 한다.

### - 시스템 구성도



## 2. 시스템 장비 구성요구사항

요구사항 고유번호		ECR-001
요구사항 명칭		웹 서버 호스팅 서비스
요구사항 분류		시스템 장비 구성 요구사항
요구사항 상세 설명	정의	웹 서버 호스팅을 위한 클라우드 서비스
	세부 내용	<ul style="list-style-type: none"> <li>- 장비 품목: Amazon EC2 인스턴스(가상 서버), Amazon Route 53 (도메인 이름)</li> <li>- 장비 수량: EC2 인스턴스 1개, Route 53 DNS 1개</li> <li>- 장비 기능: 웹 서버 개발</li> <li>- 장비 성능 및 특징: <ul style="list-style-type: none"> <li>* EC2 인스턴스: t3.medium, 2 vCPU, 4GB RAM, 50GB 스토리지, Linux/UNIX 운영체제, HTTP/HTTPS 트래픽 지원</li> <li>* Route 53 : 도메인 이름 (codepse.com), 레코드 수 5, 퍼블릭 호스팅 영역</li> </ul> </li> </ul>
요구사항 고유번호		ECR-002
요구사항 명칭		DB관리 서비스
요구사항 분류		시스템 장비 구성 요구사항
요구사항 상세 설명	정의	데이터베이스 관리 서비스
	세부 내용	<ul style="list-style-type: none"> <li>- 장비 품목: Amazon RDS(관계형 데이터베이스)</li> <li>- 장비 수량: RDS 인스턴스 1개</li> <li>- 장비 기능: 웹 서버 개발</li> <li>- 장비 성능 및 특징: <ul style="list-style-type: none"> <li>* RDS 인스턴스: db.t3.micro, MySQL 8.0 버전, 범용 SSD(gp2) 200GB 스토리지, 자동 백업 및 복원 기능 제공, 24/7 모니터링 및 알림 서비스 제공</li> </ul> </li> </ul>
요구사항 고유번호		ECR-003
요구사항 명칭		웹 프레임워크
요구사항 분류		시스템 장비 구성 요구사항
요구사항 상세 설명	정의	웹 서버 개발을 위한 프레임워크 도입
	세부 내용	<ul style="list-style-type: none"> <li>- 장비 품목: Flask 웹 프레임워크</li> <li>- 장비 수량: 1개</li> <li>- 장비 기능: 웹 서버 개발</li> <li>- 장비 성능 및 특징: Python 언어로 개발되었으며, 간편한 구성과 개발 환경을 제공함. 유연한 확장성 및 보안성이 좋음</li> </ul>



### 3. 기능 요구사항

요구사항 고유번호		SFR-001
요구사항 명칭		회원가입 및 로그인
요구사항 분류		기능
요구사항 상세 설명	정의	<ul style="list-style-type: none"> <li>- 회원가입 - 로그인 - 개인화 기능</li> <li>- 회원가입을 한 후, 로그인하여 사용자의 개인화르 보여주는 기능</li> </ul>
	세부 내용	<ul style="list-style-type: none"> <li>① 회원가입: 사용자가 서비스에 처음으로 가입할 때 사용하며, 자신의 개인 정보(예: 이름, 이메일, 비밀번호 등)를 입력하고 등록하는 기능</li> <li>② 로그인: 회원가입을 통해 등록한 회원정보를 이용하여 일반적으로 이메일 주소와 비밀번호를 사용하여 로그인하는 기능</li> <li>③ 개인화: 서비스 내에서 수행한 활동의 이력을 확인하며, 자신의 활동을 추적하고 개인적인 경험을 제공하는 기능</li> <li>④ 보안: 사용자의 개인 정보를 안전하게 보호하는 역할로, 데이터가 외부에 노출되더라도 읽을 수 없도록 보호하는 기능</li> </ul>
요구사항 고유번호		SFR-002
요구사항 명칭		코딩 테스트
요구사항 분류		기능
요구사항 상세 설명	정의	<ul style="list-style-type: none"> <li>- 코딩테스트 - 코드 제출 - 코드 피드백</li> <li>- 사용자가 입력한 코드에 대해 오류를 분석하고, 코드의 문제점을 탐지하여 개선 방안을 제시해 주는 기능</li> </ul>
	세부 내용	<ul style="list-style-type: none"> <li>① 문법적인 오류: 코드에 문법적인 오류가 있는 경우, 이를 감지하여 사용자에게 알려주는 기능</li> <li>② 코드 스타일: 코드의 가독성과 일관성을 유지하기 위해 코드 스타일 가이드에 따라 작성되었는지 검사하여 사용자에게 알려주는 기능</li> <li>③ 최적화: 입력한 코드가 더 효율적으로 동작할 수 있는 방법이 있는지 검사하여 사용자에게 추천하는 기능</li> <li>④ 보안: 입력한 코드에 보안적인 취약점이 있는지 검사하여 사용자에게 알려주는 기능</li> </ul>

요구사항 고유번호		SFR-003
요구사항 명칭		AI ChatBot
요구사항 분류		기능
요구사항 상세 설명	정의	<ul style="list-style-type: none"> <li>- 질문 - 제출 - 답변</li> <li>- 사용자가 입력한 질문에 대해 Chat GPT API를 이용하여 인공지능이 답변을 제공하는 기능</li> </ul>
	세부 내용	<ul style="list-style-type: none"> <li>① 코딩 질문 응답: 사용자가 코딩 관련 질문을 할 경우, 적절한 답변을 제공하는 기능</li> <li>② 코드 설명 및 해석: 사용자가 제출한 코드에 대한 자세한 설명과 해석을 제공하는 기능, 코드의 각 부분이 어떤 기능을 하는지 알려주는 기능</li> <li>③ 실시간 코드 디버깅: 사용자가 작성한 코드에서 발생하는 오류를 실시간으로 찾아주고, 해결 방법을 제시해 주는 기능</li> <li>④ 코드 최적화 제안 : 사용자가 작성한 코드를 분석하여, 성능을 개선할 수 있는 방법을 제시해 주는 기능</li> </ul>

요구사항 고유번호		SFR-004
요구사항 명칭		Coding Game
요구사항 분류		기능
요구사항 상세 설명	정의	<ul style="list-style-type: none"> <li>- 타이핑 게임 - 드래그 게임 - 출력값 게임</li> <li>- 기본적인 프로그래밍 언어의 흐름과 문법을 재미있고 쉽게 배울 수 있는 기능</li> </ul>
	세부 내용	<ul style="list-style-type: none"> <li>① 타이핑 게임: 정해진 문법 규칙에 따라 주어진 시간 안에 코드를 타이핑할 수 있는 기능</li> <li>② 드래그 게임: 사용자가 빈칸에 드래그하여 올바른 위치에 배치하여 문법을 배우는 기능</li> <li>③ 출력값 게임: 사용자가 주어진 코드를 해석하여 출력값을 예측하는 기능</li> <li>④ 리더보드: 사용자가 게임의 성적에 따라 리더보드에서 자신의 순위를 확인할 수 있는 기능</li> </ul>

요구사항 고유번호		SFR-005
요구사항 명칭		코드 리뷰
요구사항 분류		기능
요구사항 상세 설명	정의	<ul style="list-style-type: none"> <li>- 글 작성 - 글 수정 및 삭제 - 댓글</li> <li>- 커뮤니티 공간에서 자신의 코드를 공유하여 사용자들 간에 피드백 및 정보 공유 기능</li> </ul>
	세부 내용	<ul style="list-style-type: none"> <li>① 코드 공유: 사용자가 자신의 코드를 작성하고 설명하여 공유할 수 있는 기능</li> <li>② 글 수정: 사용자가 공유한 코드를 수정하거나 삭제할 수 있는 기능</li> <li>③ 피드백 제공: 다른 사용자들이 코드에 대한 피드백을 제공할 수 있는 기능</li> <li>④ 코드 분석 및 평가: 사용자들이 코드의 품질을 분석하거나 코드에 대한 최적화 방안을 제시하는 기능</li> </ul>

요구사항 고유번호		SFR-006
요구사항 명칭		마이페이지
요구사항 분류		기능
요구사항 상세 설명	정의	<ul style="list-style-type: none"> <li>- 회원 정보 - 코딩 테스트 이력 - 게시판 이력 - 게임 이력</li> <li>- 개인화 공간으로 자신의 활동 이력을 효과적으로 관리하고 확인하는 기능</li> </ul>
	세부 내용	<ul style="list-style-type: none"> <li>① 개인 정보 확인: 자신의 개인 정보를 확인할 수 있는 기능</li> <li>② 테스트 이력: 사용자가 자신의 코딩 테스트 이력을 조회할 수 있는 기능</li> <li>③ 게시판 이력: 작성한 글과 댓글을 확인하고 관리할 수 있는 기능</li> <li>④ 게임 이력: 플레이한 게임의 점수를 확인할 수 있는 기능</li> </ul>

#### 4. 성능 요구사항

요구사항 고유번호		PER-001
요구사항 명칭		채점 소요시간
요구사항 분류		성능
요구사항 상세 설명	정의	사용자가 코딩테스트 내에서 입력한 코드를 채점하는데 소요되는 시간
	세부 내용	- 사용자가 입력한 코드를 채점 서버로 전송하고 서버에서 해당 코드를 정답과 비교 분석하여 결과를 생성하는데, 이 결과가 다시 사용자의 화면에 정답 여부와 함께 5초 이내로 출력되어야 함.

요구사항 고유번호		PER-002
요구사항 명칭		질의응답 소요시간
요구사항 분류		성능
요구사항 상세 설명	정의	사용자가 질의응답 페이지에 입력한 질문을 서버에서 처리하고, 적절한 답변을 찾아 사용자에게 반환하는 데 걸리는 시간
	세부 내용	- 사용자가 질의응답 창에 질문을 입력하면 서버에서 해당 질문을 처리하여 3초 이내에 사용자의 화면에 답변이 출력되어야 함.



## 5. 인터페이스 요구사항

요구사항 고유번호		SIR-001
요구사항 명칭		디자인
요구사항 분류		인터페이스
요구사항 상세 설명	정의	사용자 친화적인 디자인을 적용하여, 사용자들이 웹 페이지를 보다 쉽고 편리하게 이용할 수 있도록 함.
	세부 내용	<ul style="list-style-type: none"> <li>- 웹 페이지의 색상을 조정하여 사용자 친화적인 디자인을 구성해야 함.</li> <li>- 레이아웃을 조정하여 사용자가 정보를 쉽게 파악할 수 있도록 해야 함.</li> <li>- 적절한 아이콘을 삽입하여 사용자가 직관적으로 이해할 수 있도록 해야 함.</li> <li>- 폰트를 선택하여 가독성을 높이고, 읽기 쉬운 디자인을 구성해야 함.</li> <li>- 버튼의 위치와 디자인을 고려하여 사용자가 목표를 달성하기 쉽도록 해야 함.</li> </ul>

요구사항 고유번호		SIR-002
요구사항 명칭		사용자 개별 데이터
요구사항 분류		인터페이스
요구사항 상세 설명	정의	사용자는 회원가입/로그인을 통해 자신의 개별 데이터를 관리할 수 있음.
	세부 내용	<ul style="list-style-type: none"> <li>- 사용자 정보 관리: 회원가입 시 입력한 사용자 정보(이름, 이메일, 비밀번호 등)를 관리하고 수정할 수 있도록 제공해야 함.</li> <li>- 개인 데이터 기록: 회원 가입 후에는 사용자 개인의 코딩 테스트 기록, 질의응답 기록 등을 기록하고 관리할 수 있어야 함.</li> <li>- 개인 데이터 열람: 로그인 시 개인 데이터에 접근할 수 있는 페이지가 제공되어야 함.</li> <li>- 로그아웃 기능: 사용자가 로그아웃을 클릭하면 현재 로그인된 상태가 해제되고, 다른 사용자가 로그인할 수 있도록 해야 함.</li> </ul>

## 6. 데이터 요구사항

요구사항 고유번호		DAR-001
요구사항 명칭		코딩테스트 문제
요구사항 분류		데이터
요구사항 상세 설명	정의	사용자의 코딩테스트를 위해 문제를 미리 데이터베이스에 등록
	세부 내용	<ul style="list-style-type: none"> <li>- 코딩테스트 문제를 저장할 테이블을 만들어야 하고, 테이블에는 문제 ID, 문제 제목, 문제 내용, 입력값, 출력값 등의 필드를 갖고 있어야 함.</li> <li>- 각각의 문제는 문제 ID를 가지고 구분하며, 문제 ID를 사용하여 사용자가 원하는 문제를 검색할 수 있어야 함.</li> <li>- 문제의 난이도나 유형 등 추가적인 정보도 필드로 추가할 수 있어야 함.</li> <li>- 사용자가 문제를 푸는 과정에서 입력한 값과 기대하는 출력값을 데이터베이스에 저장하여 추후 분석에 사용할 수 있음.</li> </ul>

요구사항 고유번호		DAR-002
요구사항 명칭		데이터 저장
요구사항 분류		데이터
요구사항 상세 설명	정의	사용자 개별 데이터에 대한 저장
	세부 내용	<ul style="list-style-type: none"> <li>- 회원가입 시, 입력한 사용자 정보와 비밀번호는 데이터베이스에 암호화되어 저장되어야 함.</li> <li>- 사용자가 코딩테스트를 진행하면서 입력한 코드나 결과 등의 정보도 데이터베이스에 저장되어야 함.</li> <li>- 사용자의 코드, 결과 등의 정보를 활용하여 사용자의 코딩테스트 기록과 성적을 관리함.</li> <li>- 데이터베이스에 질문과 답변 기록이 저장되어야 함.</li> </ul>

<b>요구사항 고유번호</b>		DAR-003
<b>요구사항 명칭</b>		데이터 관리
<b>요구사항 분류</b>		데이터
<b>요구사항 상세 설명</b>	<b>정의</b>	시스템에서 사용되는 데이터를 저장하고 보존하는 방안과 데이터 제약사항
	<b>세부 내용</b>	<ul style="list-style-type: none"> <li>- 데이터의 일관성을 유지할 수 있도록 데이터의 저장 방식은 일괄 저장 방식을 채택할 것이며, 데이터를 일정한 주기로 저장해야 함.</li> <li>- 데이터의 정확성을 유지하고 시스템 성능의 향상을 위해 잘못된 데이터나 중복된 데이터를 제거하고 누락된 데이터를 보완하는 데이터 정제 작업이 수행되어야 함.</li> <li>- 데이터 확장에 대한 제약 사항으로는, 시스템의 용량이 한계점에 이르렀을 때, 데이터를 분산 저장하거나, 외부 서비스를 이용할 수 있도록 구축해야 함.</li> <li>- 보안적인 요구사항으로는, 데이터의 암호화를 통해 외부로부터의 침입을 방지해야 함.</li> <li>- 데이터의 안전성을 위해 접근 권한을 제한해야 함.</li> </ul>

## 7. 테스트 요구사항

요구사항 고유번호	TER-001
요구사항 명칭	테스트 방안
요구사항 분류	테스트
요구사항 세부내용	<ul style="list-style-type: none"> <li>- 단위 테스트에서는 각 모듈에서 함수와 메서드가 예상대로 동작하는지 확인해야 함.</li> <li>- 통합 테스트에서는 각 모듈이 연동하는 부분에서 오류가 발생하지 않도록 해야 함.</li> <li>- 시스템 테스트에서는 사용자 시나리오를 통해 시스템이 예상대로 동작하는지 확인해야 함.</li> <li>- 모든 테스트는 목표 시스템이 요구하는 성능 수준을 충족시키는지 확인해야 함.</li> <li>- 테스트 환경은 최종 사용자가 실제로 사용할 환경과 유사한 환경에서 수행되어야 함.</li> <li>- 테스트 데이터는 다양한 형태의 실제 데이터의 포함해야 하고 다양한 시나리오에 대한 테스트 데이터가 충분히 제공되어야 함.</li> <li>- 테스트를 수행하기 전에 명확한 목표와 기대 결과를 설정해야 함</li> <li>- 테스트 시나리오는 실제 사용자가 기대하는 요구사항과 부합해야 함.</li> <li>- 목표 시스템이 제공할 기능 요구사항을 검증하기 위해 해당 기능에 대한 테스트 케이스를 작성하고, 이를 실행하여 예상된 결과가 도출되는지를 확인해야 함.</li> <li>- 테스트 도구 및 자동화를 사용하여 반복적인 테스트를 수행할 수 있어야 함.</li> <li>- 시스템의 기능 구현 정확성은 사용자가 직접 테스트 수행 기간에 해당 기능에 대한 테스트를 수행하여 검증함.</li> </ul>

## 8. 보안 요구사항

요구사항 고유번호		SER-001
요구사항 명칭		인터페이스 보안
요구사항 분류		보안
요구사항 상세 설명	정의	외부 시스템과의 데이터 송수신에서 데이터 암호화 및 기록을 보장하는 요구사항을 기술
	세부 내용	<ul style="list-style-type: none"> <li>- 데이터 암호화: 웹 사이트에서 전송되는 데이터는 암호화되어 전송되어야 함.</li> <li>- 개인정보 노출 방지: 웹 사이트의 화면에는 권한이나 인증 절차 없이 개인정보가 노출되지 않아야 함.</li> <li>- 취약점 방지: 웹 사이트의 인터페이스에서 발생할 수 있는 취약점(SQL Injection, XSS 공격 등)을 예방해야 함.</li> <li>- 데이터 무결성: 웹 사이트에서 전송된 데이터가 조작되지 않도록 해야 함.</li> <li>- 세션 보안: 웹 사이트에서 사용자의 세션 정보가 탈취되지 않도록 보호해야 함.</li> <li>- 보안 로그: 웹 사이트의 인터페이스에서 발생하는 모든 보안 이벤트는 로그로 기록되어야 함.</li> </ul>

요구사항 고유번호		SER-002
요구사항 명칭		네트워크 보안
요구사항 분류		보안
요구사항 상세 설명	정의	네트워크 보안 요구사항은 웹 프로젝트에서 데이터를 주고받는 네트워크를 안전하게 유지하기 위한 요구사항을 의미
	세부 내용	<ul style="list-style-type: none"> <li>- SSL 인증서를 사용하여 데이터 전송 과정에서 공격을 방지하고 방화벽을 설치하여 외부에서의 불법적인 접근을 차단해야 함.</li> <li>- 서버의 운영체제 및 네트워크 장비에 대한 보안 패치 및 설정을 지속적으로 관리해야 함.</li> <li>- 사용자 인증 및 권한 관리를 통해 시스템 접근 권한을 제한하고, 로그 기록을 철저하게 관리하여 보안 이슈가 발생했을 때 빠르게 대응할 수 있도록 준비해야 함.</li> </ul>

요구사항 고유번호		SER-003
요구사항 명칭		데이터 보안
요구사항 분류		보안
요구사항 상세 설명	정의	데이터 보안은 데이터를 보호하고 안전하게 유지 및 관리하며 데이터 무결성, 기밀성, 가용성을 보장하는 것을 의미
	세부 내용	<ul style="list-style-type: none"> <li>- 접근 권한 관리: 각 사용자의 권한에 따라 데이터에 접근할 수 있는 권한을 제한함으로써 데이터를 보호해야 함.</li> <li>- 백업과 복구: 데이터 유실이나 장애로부터 보호하기 위해 정기적으로 데이터를 백업하고 복구할 수 있는 시스템을 구축해야 함.</li> <li>- 모니터링 감시: 데이터베이스 접근 및 활동을 모니터링하여 불법적인 시도나 위험한 행위를 탐지하고 대응해야 함.</li> <li>- 업데이트 및 보안 패치: 데이터베이스 시스템과 관련된 소프트웨어 및 하드웨어의 보안 패치를 정기적으로 수행하여 보안 취약점을 최소화해야 함.</li> </ul>

요구사항 고유번호		SER-004
요구사항 명칭		로그인 보안
요구사항 분류		보안
요구사항 상세 설명	정의	사용자의 개인 정보를 안전하게 보호하며, 신뢰성 높은 로그인 시스템을 구축하여 시스템을 제공하는 것을 의미
	세부 내용	<ul style="list-style-type: none"> <li>- 이메일 검증: 사용자가 회원 가입 시 사용하는 이메일이 중복되지 않도록 검증하며, 중복 시 적절한 알림을 제공해야 함.</li> <li>- 비밀번호 암호화: 사용자의 비밀번호는 안전한 암호화 방식을 통해 데이터베이스에 저장되어야 함.</li> <li>- 비밀번호 복호화 방지: 암호화된 비밀번호는 복호화가 불가능하도록 설계해야 하며, 외부 침입 시 데이터의 유출을 최소화하여야 함.</li> </ul>

## 9. 품질 요구사항

요구사항 고유번호		QUR-001
요구사항 명칭		시스템 가용성
요구사항 분류		품질
요구사항 상세 설명	정의	시스템이 지정된 시간 동안 정상적으로 작동하며, 사용 가능한 상태를 유지할 수 있는 능력
	세부 내용	<ul style="list-style-type: none"> <li>- 시스템의 가용성은 24시간/365일 유지되어야 하며, 점검이나 유지 보수 작업등으로 인한 시스템의 정상적인 운영이 중지되는 시간은 최소화되어야 함.</li> <li>- 시스템의 정상적인 운영이 중지되었을 경우, 복구 시간은 30분 이내로 유지되어야 함.</li> <li>- 유지 보수 작업, 서버 문제 등으로 시스템 운영 중지가 예상되는 경우 사용자에게 사전 공지해야 함.</li> </ul>

요구사항 고유번호		QUR-002
요구사항 명칭		사용자 편의성
요구사항 분류		품질
요구사항 상세 설명	정의	서비스 또는 시스템을 사용하는 사용자가 쉽게 이해하고 사용할 수 있도록 설계, 개발 및 제공하는 정도
	세부 내용	<ul style="list-style-type: none"> <li>- 사용자 인터페이스(UI)는 직관적이고 사용하기 쉬워야 함,</li> <li>- 시스템 기능은 명확하고 일관성 있게 제공되어야 함.</li> <li>- 시스템을 사용하는 과정에서 사용자가 경험할 수 있는 불만족 요소를 최소화해야 함.</li> </ul>

## 10. 제약 사항

요구사항 고유번호	COR-001
요구사항 명칭	시스템 개발 제약 사항
요구사항 분류	제약 사항
요구사항 세부내용	<ul style="list-style-type: none"> <li>- 시스템 개발에 사용될 프레임워크, 프로그래밍 언어, 개발 방법론 등은 조직에서 사용 가능한 범위 내에서 선택되어야 함.</li> <li>- 시스템 개발 시 개발 생산성과 유지 보수성을 고려하여 적절한 개발 프레임워크 및 개발 도구를 사용해야 함.</li> <li>- 시스템 개발 시 개발 단계마다 적절한 테스트 및 검증 절차를 수행하여 개발 중 발생 가능한 결함 및 오류를 최소화해야 함.</li> <li>- 시스템 개발 시 향후 확장성과 유지 보수성을 고려하여 모듈화 및 표준화를 적용한 개발이 이루어져야 함.</li> </ul>

요구사항 고유번호	COR-002
요구사항 명칭	시스템 설계 및 구현 제약 사항
요구사항 분류	제약 사항
요구사항 세부내용	<ul style="list-style-type: none"> <li>- 시스템 설계 시 전체적인 시스템 구조를 고려하여 모듈화 및 계층화가 이루어져야 함.</li> <li>- 시스템 구현 시 적절한 개발 프레임워크 및 개발 도구를 사용하여 개발이 이루어져야 함.</li> <li>- 시스템 구현 시 모듈화 및 표준화를 적용하여 유지 보수성 및 확장성을 고려한 구현이 이루어져야 함.</li> </ul>



## 11. 프로젝트 관리 요구사항

요구사항 고유번호	PMR-001
요구사항 명칭	품질관리
요구사항 분류	프로젝트 수행 조직
요구사항 세부내용	<ul style="list-style-type: none"> <li>- 프로젝트의 수행 조직 구성 및 역할, 책임, 권한 등을 명확히 제시해야 함.</li> <li>- 프로젝트 수행 조직 간의 의사소통 및 협업을 원활하게 할 수 있는 방안을 제시해야 함.</li> <li>- 프로젝트 수행을 위한 인력, 예산 등의 자원을 적절히 관리할 수 있는 방안을 제시해야 함.</li> </ul>

요구사항 고유번호	PMR-002
요구사항 명칭	품질관리
요구사항 분류	프로젝트 관리
요구사항 세부내용	<ul style="list-style-type: none"> <li>- 프로젝트 일정을 적절한 수준에서 세분화하여 일정 조정을 할 수 있도록 방안을 제시해야 함.</li> <li>- 일정 계획에 따른 프로젝트 수행 상황을 측정하고 분석하는 방법을 제시해야 함.</li> </ul>