프로젝트명 : 초보 클라이머를 위한 자세 추정 기반 분석 플랫폼

캡스톤 디자인II, 중간보고서

Version 1.0

개발 팀원 명(팀리더):구남석
대표 연락처:010-9475-2931
e-mail: 20201047@edu.hanbat.ac.kr

캡스톤 디자인 II 중간보고서 내용

1. 요구사항 정의서에 명세된 기능에 대하여 현재까지 진척된 결과 및 그 내용을 기술하시오.
현재 프로젝트의 핵심 기능인 영상 분석 프로세스가 구현 완료되었으며, 사용자는 클라이언트(웹)를 통해 영상을 업로드하고, 로컬 서버에서 해당 영상을 분석 및 시각화하여, 최종 결과물을 다시 클라이언트로 전송해 재생하는 전체 흐름이 정상적으로 동작함을 확인하였습니다.
실제 서버 구현 및 회원 정보 관리를 위한 데이터베이스 구축이 필요한 상태입니다.

1.1. 소스코드 일부
[백엔드 - app.py]
서버의 핵심 로직은 Flask 프레임워크를 기반으로 작성되었습니다. 아래 코드는 클라이언트의 영상 업로드 요청(POST /analyze)을 처리하고, 분석이 완료된 영상의 경로를 JSON으로 반환하는 API 엔드포인트 부분입니다.

```
# app.py 일부
# ... (영상 분석 함수 analyze_video 정의) ...
# ================== API 라우트 ==================
@app.route('/analyze', methods=['POST'])
def analyze():
    """ 클라이언트로부터 영상 파일을 받아 분석을 수행하고 결과 경로를 반환하는 API """
    if 'video' not in request.files:
        return jsonify({'error': 'No video uploaded'}), 400
    video_file = request.files['video']
    # ... (파일 저장 및 경로 설정) ...
    try:
        # 핵심 분석 로직 호출
        analyze_video(video_path, result_path)
    except Exception as e:
        # 오류 발생 시 500 에러 반환
        return jsonify({'error': str(e)}), 500
    # 성공 시, 결과 영상의 경로를 JSON으로 응답
    return jsonify({'result_video': f'/results/{result_filename}'})
@app.route('/results/<filename>')
def get_result(filename):
    """ 클라이언트가 요청한 결과 영상 파일을 실제로 전송하는 API """
    return send_from_directory(app.config['RESULT_FOLDER'], filename)
```

[프론트엔드 - api_service.dart]
Flutter 클라이언트가 Flask 서버와 통신하는 로직을 담당하는 서비스 모듈입니다. http 패키지를 사용하여 서버의 /analyze 엔드포인트로 영상 파일을 POST 방식으로 전송합니다

```
// services/api_service.dart 일부
import 'package:http/http.dart' as http;
import 'dart:io' show File;
class ApiService {
  static const String serverUrl = 'http://127.0.0.1:5000';
  // Flutter 모바일용 업로드
  static Future<void> uploadVideoMobile(File file) async {
    final uri = Uri.parse('$serverUrl/analyze');
    final request = http.MultipartRequest('POST', uri);
    request.files.add(await http.MultipartFile.fromPath('video', file.path));
    final response = await request.send();
    if (response.statusCode != 200) {
      final respText = await response.stream.bytesToString();
      throw Exception('Mobile video upload failed: $respText');
    }
  }
  // ... (Web 전용 업로드 로직) ...
}
```

[프론트엔드 - video_upload_screen.dart]
사용자 인터페이스(UI)와 상태 관리를 담당하는 스크린 위젯입니다. 사용자가 'Select Video' 버튼을 눌렀을 때 _pickFile 함수가 호출되어 ApiService를 통해 서버에 영상을 업로드하고, 로딩 상태를 관리하며, 최종적으로 반환된 영상 경로를 비디오 플레이어로 재생합니다.

```
// screens/video_upload_screen.dart 일부
class _VideoUploadScreenState extends State<VideoUploadScreen> {
  // ... (상태 변수 선언: selectedFileName, isUploading 등) ...
  Future<void> _pickFile() async {
    // ... (웹/모바일 환경에 따라 파일 선택) ...
    if (videoFile != null) {
      setState(() {
        selectedFileName = videoFile.path.split('/').last;
        isUploading = true; // 업로드 시작 -> 로딩 UI 표시
        resultVideoPath = null;
      });
      try {
        // ApiService를 통해 서버에 분석 요청
        await ApiService.uploadVideoMobile(videoFile);
        if (!mounted) return;
        // 서버로부터 받은 경로로 결과 영상 URL 구성
        String mp4Name = selectedFileName!.split('.').first + ".mp4";
        resultVideoPath = "${ApiService.serverUrl}/results/result_$mp4Name";
        isUploading = false; // 업로드 종료
        // 비디오 플레이어 초기화 및 재생
        _initializeVideoPlayerMobile(resultVideoPath!);
      } catch (e) {
        // ... (에러 처리) ...
      }
    }
  }
}
```

현재 버전(v1.0)은 로컬 서버 환경에서 실행됩니다. 실행을 위해서는 아래의 환경 구성이 필요합니다.
언어 및 라이브러리: Python 3.11, Flask, OpenCV, MediaPipe 등 (requirements.txt 기반 설치)

```
flask==3.0.3
flask-cors==5.0.0
opencv-python==4.10.0.84
mediapipe==0.10.14
numpy==1.26.4
```

외부 도구: FFmpeg (영상 회전 보정 및 인코딩용)
실행: 터미널에서 python app.py 명령어를 통해 Flask 개발 서버를 실행합니다.
클라이언트: Flutter 웹 개발 서버를 실행하여 localhost로 접속합니다.

1.2. 소프트웨어 아키텍처

본 시스템은 역할 분리를 위해 클라이언트-서버 아키텍처로 설계되었습니다.

Client (Flutter Web/Mobile): 사용자 인터페이스(UI)와 사용자 경험(UX). 서버에 영상 분석을 요청하고, 최종 결과물을 받아와 사용자에게 보여주는 역할만 수행합니다.

Server (Python Flask): AI 연산과 비즈니스 로직을 처리.

API Gateway: Flask가 /analyze, /results 등의 API 엔드포인트를 통해 클라이언트의 요청을 접수.

Preprocessing: FFmpeg를 이용해 영상 회전 문제를 자동 보정.

Core Analysis: MediaPipe로 자세를 추정하고, OpenCV로 결과를 시각화.

Postprocessing: FFmpeg로 최종 영상을 웹 표준(H.264)으로 인코딩.

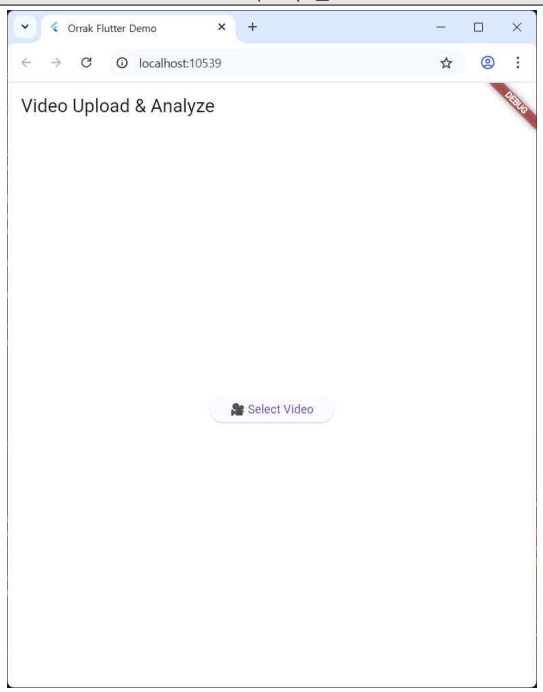
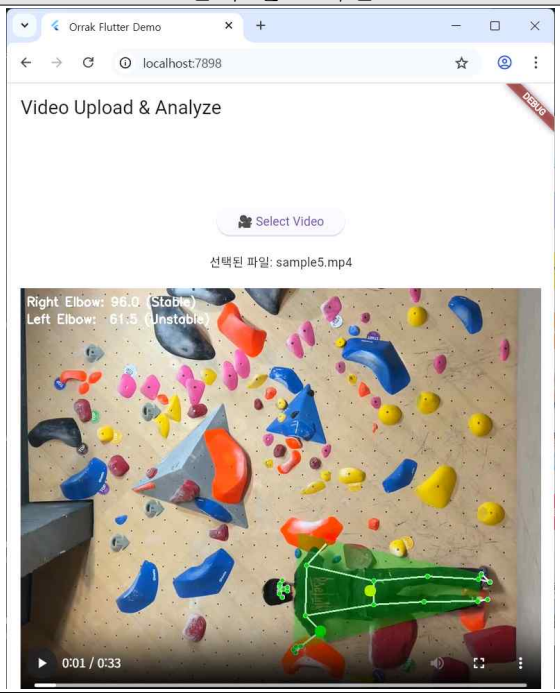
1.3. 기능별 상세 요구사항

현재 구현된 핵심 기능의 유스케이스는 다음과 같습니다.

- 유스케이스명: 비회원 영상 자세 분석
- 액터: 사용자
- 사전 조건: 사용자는 분석을 원하는 클라이밍 영상 파일을 가지고 있다.
- 시나리오:
    1. 사용자가 서비스 페이지에 접속한다.
    2. 사용자가 '영상 선택' 버튼을 클릭하여 로컬 파일을 선택한다.
    3. 시스템은 선택된 영상을 서버로 업로드하고, 분석이 진행 중임을 사용자에게 표시한다.
    4. 서버는 영상 분석 및 시각화 작업을 완료한다.
    5. 시스템은 분석이 완료된 영상을 클라이언트의 비디오 플레이어를 통해 재생한다.
    6. 사용자가 분석 결과를 확인한다.

1.4. UI 프로토타입

UI는 사용자가 최소한의 동작으로 핵심 기능을 이용할 수 있도록 직관적으로 설계되었습니다. 아래는 현재 개발 완료된 핵심 페이지의 프로토타입입니다.

초기 화면	분석 완료 화면
	

1.5. DB 설계 모델

현재 진행 상황은 서버의 자세 분석 요청 및 응답에 초점을 맞추어 진행했습니다. 따라서, 사용자 관리, 영상 기록 관련한 데이터베이스는 설계되지 않았으며, 다음 마일스톤에서 개발될 예정입니다.

2. 프로젝트 수행을 위해 적용된 추진전략, 수행 방법의 결과를 작성하고, 만일 적용과정에서 문제점이 도출되었다면 그 문제를 분석하고 해결방안을 기술하시오.

프로젝트는 '핵심 기능 우선 구현' 전략을 채택하여, 가장 중요한 영상 분석 파이프라인을 먼저 구축하는 데 집중하였다. 이를 통해 조기에 기술적 타당성을 검증하고 실제 동작하는 결과물을 확보할 수 있었다.

2.1. 문제 해결 과정 및 결과

문제점	문제 분석	해결 방안
1. 영상 업로드 시 회전 문제	영상의 해상도(가로/세로 길이)를 비교하여 회전 여부를 판단했으나, 일부 기기에서 메타데이터만 변경하고 해상도는 그대로 저장하여 실패	OpenCV로 영상을 읽기 전, FFmpeg를 이용해 영상의 회전 메타데이터를 강제로 적용하는 전처리 단계를 추가하여 모든 영상이 정방향으로 보정되도록 문제를 해결
2. 긴 분석 시간 및 서버 부하	고해상도 영상 분석 시 2-3분의 긴 처리 시간 소요. 현재의 동기 방식으로는 동시 접속자 증가 시 서버가 다운될 위험이 존재	비동기 작업 큐(Job Queue)를 도입하여 사용자의 요청을 대기열에 등록하고, 서버는 순차적으로 처리 후 완료 시 클라이언트에게 알림을 주는 방식으로 아키텍처 개선 필요
3. 백엔드 코드의 복잡성 증가	빠른 프로토타이핑을 위해 모든 서버 로직(API, 분석, 전/후처리)이 app.py 단일 파일에 집중되어 유지보수성이 저하	향후 리팩토링(Refactoring)을 통해 코드를 기능별로 모듈화할 계획

2.2. 프로젝트 일정 계획 준수 현황

9월 5일 현재, '2단계: 핵심 기능 개발' 마일스톤(9월 12일)을 앞두고 목표했던 End-to-End 프로토타입 구현을 완료하여 일정을 준수하고 있습니다. '회원 관리' 기능은 핵심 기능의 기술적 난이도와 안정성 확보를 위해 전략적으로 다음 단계로 범위를 조정했습니다. 남은 기간 동안 테스트 및 안정화 작업을 거쳐 3단계로 넘어갈 계획입니다.

[별첨1]

프로젝트명 : 초보 클라이머를 위한 자세 추정 기반 분석 플랫폼

소프트웨어 요구사항 정의서

Version 1.0

개발 팀원 명(팀리더):구남석
대표 연락처:010-9475-2931
e-mail: 20201047@edu.hanbat.ac.kr

목차

요구사항 정의서에 사용되는 양식 설명

요구사항 고유번호(ID): 제안요청서에 정의된 요구사항에 대해 계약, 사업수행, 사업완료 및 검수까지 변경, 삭제, 수정 여부에 대한 추적관리를 위해 고유의 번호를 부여하도록 한다.

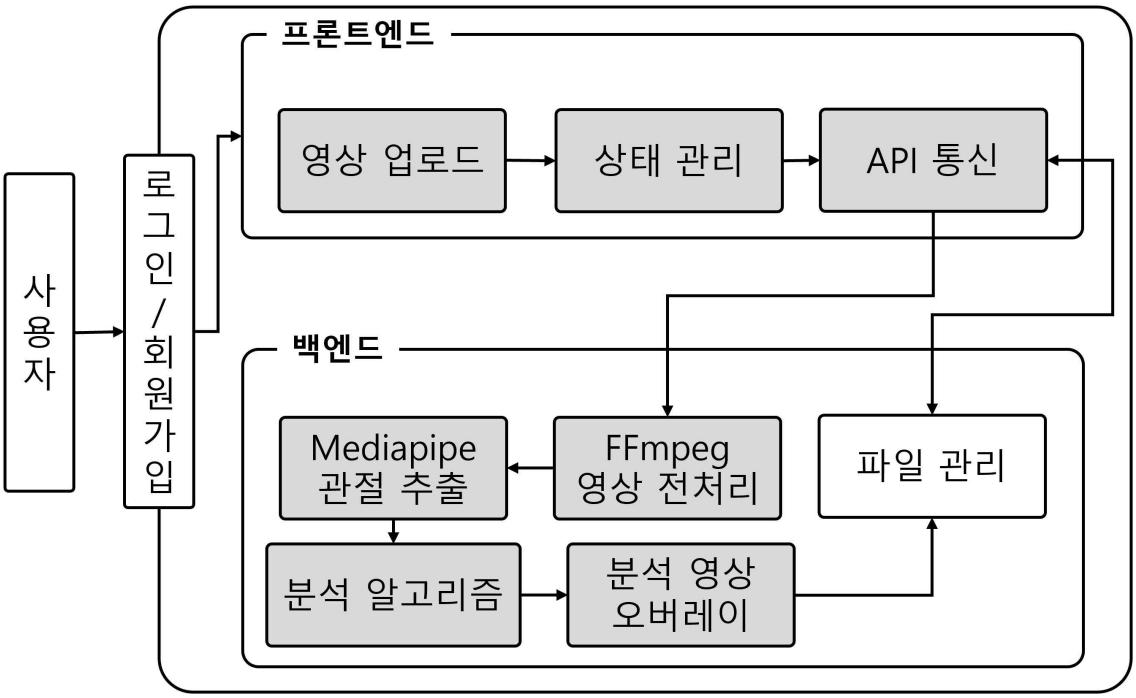
요구사항 구분 및 ID부여 규칙

요구사항 구분		ID 부여 규칙
시스템 장비 구성 요구사항	Equipment Composition Requirement	ECR-000
기능 요구사항	System Function Requirement	SFR-000
성능 요구사항	Performance Requirement	PER-000
인터페이스 요구사항	System Interface Requirement	SIR-000
데이터 요구사항	Data Requirement	DAR-000
테스트 요구사항	Test Requirement	TER-000
보안 요구사항	Security Requirement	SER-000
품질요구사항	Quality Requirement	QUR-000
제약사항	Constraint Requirement	COR-000
프로젝트 관리 요구사항	Project Mgmt. Requirement	PMR-000
프로젝트 지원 요구사항	Project Support Requirement	PSR-000

요구사항 세부내용 작성표 양식 및 항목설명

요구사항 고유번호		(설명) 요구사항 추적관리를 위해 독립적인 고유번호(ID) 부여
요구사항 명칭		(설명) 요구사항 명칭을 작성함
요구사항 분류		(설명) 요구사항 분류기준에 따른 분류를 기입
요구사항 상세 설명	**정의**	(설명) 요구사항 정의
	세부 내용	(설명) 요구사항 구체적인 세부 내용을 설명
산출정보		(설명) 해당기능을 통해 산출되는 결과물 혹은 정보를 표기
관련 요구사항		(설명) 정의된 요구사항과 관련된 요구사항에 대해 기술
요구사항 출처		(설명) 기능 도출내용에 대한 출처(source) 표기

1. 시스템 개요



2. 시스템 장비 구성요구사항

요구사항 고유번호		ECR-001		
요구사항 명칭		사용자 디바이스 최소 권장 사양		
요구사항 분류		시스템 장비 구성 요구사항	**응락수준**	필수
요구사항 상세설명	정의	일반 사용자가 보유한 스마트폰·노트북에서 영상 업로드 및 분석 결과 확인이 가능하도록 현실적인 최소·권장 사양을 정의		
	세부 내용	- 최소: Android 9 / iOS 13 이상 - 권장: 스마트폰 RAM ≥ 4GB, Android 11+ /iOS14+, 노트북 쿼드코어·RAM ≥ 8GB, 업로드 5 Mbps 이상.		

요구사항 고유번호		ECR-002		
요구사항 명칭		서버(분석·인코딩) 최소·권장 하드웨어 사양		
요구사항 분류		시스템 장비구성 요구사항	**응락수준**	필수
요구사항 상세설명	정의	입력되는 일반 화질(720p~1080p) 영상을 안정적으로 전처리·AI 분석·인코딩할 수 있는 서버 하드웨어 요구사항을 정의		
	세부 내용	- 최소(개발/테스트): CPU: 쿼드코어(4코어)를 지원하는 Intel i3 / AMD Ryzen 3 (최근 5~6년 내 모델), RAM: 8GB, FFmpeg 실행 환경 필요.		

3. 기능 요구사항

요구사항 고유번호		SFR-001		
요구사항 명칭		영상 업로드 기능		
요구사항 분류		기능	**응락수준**	필수
요구사항 상세설명	**정의**	사용자가 자신의 클라이밍 영상을 플랫폼에 업로드할 수 있는 기능		
	세부 내용	- 브라우저/앱에서 파일 선택·드래그앤드롭·카메라 직접 촬영을 지원 - 업로드 중 진행률 표시 - 파일 포맷(MP4/H.264 권장)·최대파일크기 제한(예: 40MB 기본) - 업로드 실패 시 재시도/재개 지원		

요구사항 고유번호		SFR-002		
요구사항 명칭		업로드 상태 및 처리 대기열 표시		
요구사항 분류		기능	**응락수준**	필수
요구사항 상세설명	**정의**	업로드 이후 영상의 서버 처리 상태(대기, 처리중, 완료, 실패)를 사용자에게 실시간으로 제공		
	세부 내용	- 작업 ID 발급 - 실시간 상태 폴링 또는 웹소켓 기반 푸시 - 예상 처리시간/진행률 표기(가능한 경우) - 오류 발생 시 에러 코드·원인 메시지 제공 및 재요청 버튼 제공		

요구사항 고유번호		SFR-003		
요구사항 명칭		영상 전처리(정방향·리사이즈·포맷 변환)		
요구사항 분류		기능	**응락수준**	필수
요구사항 상세설명	**정의**	입력된 영상을 분석에 적합한 형식으로 자동 전처리		
	세부 내용	- 회전 메타데이터 자동 교정 - 해상도 리사이즈 - 코덱/컨테이너 변환(FFmpeg)		

요구사항 고유번호		SFR-004		
요구사항 명칭		AI 자세 분석(관절 추적 및 지표 산출)		
요구사항 분류		기능	응락수준	필수
요구사항 상세설명	정의	영상 프레임별로 신체 관절을 추적하고 자세 관련 지표(팔꿈치 각도, COM, BOS 등)를 산출		
	세부 내용	- MediaPipe 기반 관절 33개 포인트 추출 - 프레임별 관절 좌표 저장 및 팔꿈치/어깨/무릎 각도 계산 모듈 - 무게중심(CoM) 추정 알고리즘 - 안정성(BOS) 판단 로직, 분석 결과의 신뢰도(Confidence) 수치 포함.		

요구사항 고유번호		SFR-005		
요구사항 명칭		분석 결과 시각화 및 오버레이 생성		
요구사항 분류		기능	응락수준	필수
요구사항 상세설명	정의	산출된 분석 데이터를 원본 영상에 오버레이하여 시각적으로 표시		
	세부 내용	- 스켈레톤·관절 포인트·무게중심 마커·BOS 다각형 오버레이 - 각 프레임별 텍스트(예: 팔꿈치 각도, 안정/불안정 표시) 삽입 - 하이라이트(문제 프레임 표시) 기능 - 시각화된 결과를 포함한 재생 가능한 MP4 출력		

요구사항 고유번호		SFR-005		
요구사항 명칭		결과 재생		
요구사항 분류		기능	응락수준	필수
요구사항 상세설명	정의	사용자가 분석 완료 후 결과 영상을 재생 및 다운로드하고, 과거 분석 이력을 조회 관리		
	세부 내용	- 웹/앱 내 비디오 플레이어(오버레이 토글 기능 포함) - 결과 영상 다운로드 - 사용자 요청 시 원본/결과 삭제 옵션		

4. 성능 요구사항

요구사항 고유번호		PER-001		
요구사항 명칭		영상 처리 지연(단일 세션)		
요구사항 분류		성능	**응락수준**	필수
요구사항 상세설명	**정의**	서버가 단일 업로드 영상(720p, 30s)을 전처리·AI분석·시각화·인코딩까지 완료하는 데 걸리는 목표 지연시간을 정의		
	세부 내용	- 목표(개발/테스트 서버, CPU 기반): 30초 길이 720p 영상 → 전체 처리시간 ≤ 180초. - 목표(권장 사양/GPU 사용 시): 동일 영상 → 전체 처리시간 ≤ 90초.		

요구사항 고유번호		PER-002		
요구사항 명칭		동시 처리 용량		
요구사항 분류		성능	**응락수준**	필수
요구사항 상세설명	**정의**	시스템이 동시 업로드·처리할 수 있는 세션 수 및 확장 정책을 정의		
	세부 내용	- 초기(단일 권장 서버): 최소 10개의 동시 분석 작업(대기 포함)을 유지할 수 있어야 함(각 작업은 PER-001 목표를 기준으로 스케줄링). - 확장: Kubernetes 등으로 수평 확장 시 동시 세션 수는 노드 추가에 비례해 선형 확장		

5. 인터페이스 요구사항

요구사항 고유번호		SIR-001		
요구사항 명칭		영상 업로드(Analyze) API 규격		
요구사항 분류		인터페이스	**응락수준**	필수
요구사항 상세설명	정의	클라이언트가 영상 파일을 서버로 전송하여 분석을 요청하는 REST 엔드포인트의 요청/응답 규격을 정의		
	세부 내용	- 요청: POST /api/v1/analyze (멀티파트/form-data) — 필드: video(파일), user_id(string), metadata(JSON: 촬영일시/기기/해상도/프레임) - 응답: 업로드 수신 확인 + job_id, status: accepted, estimated_wait_seconds - 인증: Bearer 토큰(Authorization header) 또는 세션 쿠키 필요		
기타 고려 사항		- 대용량/불안정 네트워크 대비 resumable upload(Chunked 또는 tus/s3 multipart) - 권장; HTTPS 필수; 클라이언트 재시도 로직과 업로드 진행률 표시 필요. (현재 샘플 구현은 /analyze 엔드포인트를 사용함)		
산출 정보		산출정보: 수신확인 JSON(예: {job_id, status}), 업로드 실패 에러코드 목록		
요구사항 출처				

요구사항 고유번호		SIR-001		
요구사항 명칭		결과 조회 및 다운로드 API 규격		
요구사항 분류		인터페이스	**응락수준**	필수
요구사항 상세설명	정의	분석 완료 후 시각화된 결과 비디오, 오버레이 데이터 및 요약 리포트를 조회·다운로드 할 수 있는 규격을 정의		
	세부 내용	- 엔드포인트: GET /api/v1/results/{job_id} → 반환: { summary, metrics_url, overlay_url, result_video_url, report_url } - 파일 접근: presigned URL 방식(단기 만료) 또는 인증된 스트리밍 엔드포인트		
기타 고려 사항				
산출 정보		JSON 지표(프레임별 confidence 포함), 오버레이 비디오 URL, 최종 MP4(시각화 포함)		
요구사항 출처				

6. 데이터 요구사항

요구사항 고유번호	DAR-001		
요구사항 명칭	입력 영상 데이터 명세		
요구사항 분류	데이터	응락수준	필수
요구사항 세부내용	- 파일 형식: MP4, MOV 등 웹 표준 비디오 형식을 지원(권장: MP4) - 파일 크기: 업로드 가능한 파일의 최대 크기를 100MB로 제한 - 영상 해상도: 자세 추정을 위해 최소 HD(1280x720) 이상 해상도 - 영상 내용: 영상에는 분석 대상이 되는 클라이머 1인이 정면으로 명확하게 보여야 함		

요구사항 고유번호	DAR-002		
요구사항 명칭	자세 랜드마크 데이터 명세		
요구사항 분류	데이터	응락수준	필수
요구사항 세부내용	- 데이터 구조: 각 랜드마크는 4개의 부동소수점 값 (x, y, z, visibility)으로 구성 - 좌표계: x, y는 프레임의 너비와 높이에 대한 상대 좌표(0.0 ~ 1.0)이며, z는 카메라로부터의 추정 깊이 값 -가시성: visibility는 해당 랜드마크가 프레임 내에서 보이는지에 대한 신뢰도(0.0 ~ 1.0)를 나타냄 - 데이터 형식: 분석 과정 중에는 메모리 내 객체로 처리되며, 향후 로그 기록이나 디버깅을 위해 JSON 형식으로 직렬화(serialize)될 수 있어야 함		

요구사항 고유번호	DAR-003		
요구사항 명칭	최종 분석 영상 데이터 명세		
요구사항 분류	데이터	응락수준	필수
요구사항 세부내용	- 컨테이너 형식: MP4 - 비디오 코덱: H.264 (libx264). 웹 및 모바일 환경에서의 표준 호환성을 보장 - 오디오 코덱: AAC. 원본 영상의 오디오를 유지하며, 비디오 코덱과의 호환성을 보장		

7. 테스트 요구사항

요구사항 고유번호	TER-001		
요구사항 명칭	백엔드 단위 테스트		
요구사항 분류	테스트	응락수준	필수
요구사항 세부내용	- 테스트 대상: 영상 처리 모듈: analyze_video 함수 내 영상 회전, 인코딩 로직 - 분석 알고리즘 모듈: calculate_angle, compute_total_com 등 핵심 계산 함수 - 테스트 환경: Python의 pytest와 같은 표준 테스트 프레임워크를 사용 - 테스트 방법: 미리 정의된 입력값(샘플 랜드마크 데이터 등)에 대해 예상된 출력값이 반환되는지 확인하는 방식으로 진행		

8. 보안 요구사항

요구사항 고유번호	SER-001		
요구사항 명칭	서버 보안 강화		
요구사항 분류	보안	응락수준	필수
요구사항 세부내용	- 운영 환경에서는 Flask 서버의 디버그 모드를 반드시 비활성화하여, 오류 발생 시 민감한 시스템 내부 정보가 노출되지 않도록 함 - FFmpeg와 같은 외부 프로세스 호출 시, 사용자 입력값이 명령어의 일부로 직접 사용되지 않도록 하여 커맨드 인젝션 공격을 방지 - pip, npm 등 패키지 매니저를 통해 설치된 모든 외부 라이브러리(종속성)는 정기적으로 최신 버전으로 업데이트하여 알려진 보안 취약점을 패치		

요구사항 고유번호	SER-002		
요구사항 명칭	데이터 보관 및 파기 정책		
요구사항 분류	보안	응락수준	필수
요구사항 세부내용	- 사용자가 업로드한 원본 영상과 분석 과정에서 생성된 모든 임시 파일은 최종 결과 영상이 생성된 후 즉시 서버에서 물리적으로 삭제 - 최종 분석 결과 영상은 사용자에게 제공된 후, 사전에 정의된 보관 기간(예: 24시간)이 지나면 자동으로 파기		

9. 품질 요구사항

요구사항 고유번호	QUR-001		
요구사항 명칭	분석 정확도		
요구사항 분류	품질	응락수준	필수
요구사항 세부내용	- 랜드마크 안정성: 동일한 영상을 반복적으로 분석했을 때, 각 프레임의 랜드마크 좌표값 오차는 5% 미만 - 일관성: 랜드마크 안정성 기준을 만족하는 조건에서, 계산된 팔꿈치 각도 및 무게중심 좌표값의 오차 또한 5% 미만 - 정성적 평가: 전문가(클라이밍 코치 등)가 육안으로 확인했을 때, 분석 결과가 실제 인간의 움직임과 명백히 다른 비상식적인 결과를 출력해서는 안 됨		

요구사항 고유번호	QUR-002		
요구사항 명칭	사용성		
요구사항 분류	품질	응락수준	선택
요구사항 세부내용	- 작업 완료 시간: 처음 사용하는 사용자 기준으로, 영상을 선택하고 업로드하여 분석을 시작하기까지의 과정이 10초 이내에 완료될 수 있도록 UI/UX를 설계 - 오류 메시지 명확성: 파일 업로드 실패, 서버 통신 오류 등 문제 발생 시, 사용자가 상황을 명확히 인지하고 다음 행동을 취할 수 있도록 이해하기 쉬운 오류 메시지를 제공 - 결과 해석 용이성: 분석 결과 영상에 표시되는 시각적 정보(스켈레톤, 마커, 텍스트)는 사용자가 직관적으로 자신의 자세를 이해하는 데 도움을 제공		

요구사항 고유번호	QUR-003		
요구사항 명칭	가용성		
요구사항 분류	품질	응락수준	선택
요구사항 세부내용	- 서버의 CPU, 메모리 사용량 등 주요 자원에 대한 모니터링 시스템을 구축하여 장애 발생 가능성을 사전에 감지하고 대응 - 시스템 업데이트 및 점검은 서비스 이용자가 가장 적은 시간대(예: 새벽)에 수행하는 것을 원칙		

요구사항 고유번호	QUR-003		
요구사항 명칭	유지보수성		
요구사항 분류	품질	응락수준	선택
요구사항 세부내용	- 모듈성: 백엔드 코드는 기능(API 라우팅, 분석 서비스, 유틸리티 등)에 따라 명확하게 분리된 모듈 구조를 가짐 - 문서화: 주요 API의 기능, 요청/응답 형식 등은 Swagger와 같은 도구를 활용하여 자동으로 문서화 필요		

10. 제약 사항

요구사항 고유번호	COR-001		
요구사항 명칭	개인정보보호법 준수		
요구사항 분류	제약사항	응락수준	필수
요구사항 세부내용	- 사용자로부터 영상 데이터 수집 및 이용에 대한 명시적인 동의를 받아야 함 - 사용자로부터 영상 데이터 수집 및 이용에 대한 명시적인 동의를 받아야 한다. 수집된 데이터는 분석 목적 이외로 사용될 수 없으며, 제3자에게 제공할 수 없음 - 데이터 보관 및 파기 정책(SER-002)을 수립하고, 이를 개인정보처리방침에 명시하여 사용자에게 고지해야 하며 법률적 요구사항 변경 시, 시스템의 데이터 처리 정책을 즉시 반영		

요구사항 고유번호	COR-002		
요구사항 명칭	기술 스택 및 라이브러리		
요구사항 분류	제약사항	응락수준	필수
요구사항 세부내용	- Python 3.11, Flask 웹 프레임워크 - 프론트엔드 Dart 언어, Flutter 프레임워크 - 핵심 라이브러리: 자세 추정: Google MediaPipe 영상 처리: OpenCV, FFmpeg - 명시된 기술 스택 이외의 기술을 도입할 경우, 아키텍처 호환성 및 라이선스 검토가 반드시 선행		

요구사항 고유번호	COR-003		
요구사항 명칭	개인정보 보호 및 법규 준수 제약		
요구사항 분류	제약사항	응락수준	필수
요구사항 세부내용	- 개인정보 보호법에 따른 데이터 암호화·익명화 처리 - 대학 네트워크 보안 정책(포트·프로토콜) 준수 - 로그 보관 및 열람 권한 절차 수립		

11. 프로젝트 관리 요구사항

요구사항 고유번호	PMR-001		
요구사항 명칭	프로젝트 일정 계획 및 WBS		
요구사항 분류	품질관리	응락수준	필수
요구사항 세부내용	- 전체 프로젝트 기간: 2025년 8월 4일 ~ 2025년 10월 10일 - 1단계 (분석/설계): ~ 8월 16일 (요구사항 정의, 아키텍처 설계) - 2단계 (핵심 기능 개발): ~ 9월 12일 (백엔드 분석 모듈, 클라이언트 UI 개발) - 3단계 (통합 및 테스트): ~ 9월 26일 (클라이언트-서버 연동, 단위/통합 테스트) - 4단계 (안정화 및 배포): ~ 10월 10일 (최종 QA, 버그 수정, 시스템 배포)		

요구사항 고유번호	PMR-002		
요구사항 명칭	산출물 관리		
요구사항 분류	품질관리	응락수준	필수
요구사항 세부내용	- 버전 관리 시스템: 모든 소스코드는 Git을 사용하여 버전을 관리 - 중앙 저장소: GitHub를 중앙 소스코드 저장소로 사용 - 문서 관리: 요구사항 정의서, 설계서 등 주요 산출물은 Github 및 Google Drive 을 통해 관리		