Innovation Academy

「2025 IA x AI 해커톤」

개발 완료 보고서

팀　　명 : Baro(바로)

프로젝트명 : 포슬리(Posely)

※ 유의사항

1. 본 보고서의 내용은 최대 2 page이내로 작성 (본 표지 제외)
2. 보고서의 설명을 보충하기 위해 필요한 **사진 또는 그래프 첨부** 가능
3. 제출 서류는 일체 반환을 하지 않음
4. 제출 파일명 작성 요령
   - 파일명: [2025 IA x AI 해커톤]_팀명
5. 서체: 맑은고딕, 크기: 12p, 줄간격: 160%
6. 제출처: 깃허브에 업로드

Innovation Academy

「2025 IA x AI 해커톤」

※ 개발물의 내용을 2쪽 이내 분량으로 작성(분량 초과 시, 붙임으로 작성)

프로젝트명	포슬리(Posely)
프로젝트 목표	본 프로젝트는 AI 기반 컴퓨터 비전 기술을 활용하여 장시간 컴퓨터 사용 중 발생하는 거북목 자세를 실시간으로 감지하고 교정 피드백을 제공하는 온디바이스 애플리케이션을 개발하는 것을 목표로 한다. 사용자는 웹캠만으로 자신의 자세를 확인할 수 있으며, 데이터는 모두 로컬 환경(SQLite)에 저장되어 개인 정보 유출 위험이 없다. **기대효과** - 잘못된 자세 습관을 교정하여 목·어깨 통증 완화 및 집중력 향상 - 외부 서버 없이 온디바이스에서 동작하여 개인정보 보호 강화 - 추가 장비 없이 저비용으로 자세 관리 가능 - 향후 스트레칭 가이드나 개인 맞춤 리포트 기능으로 확장 가능
개발 환경	- 개발 언어 : TypeScript, JavaScript - 프레임워크 : Electron (데스크톱 환경), React - AI 모델 : MediaPipe Pose - 데이터베이스 : SQLite 데이터 처리 : MediaPipe API - 버전 관리 : GitHub
구현 기능	**실시간 자세 감지** - Electron 환경에서 웹캠 영상을 받아 렌더링하고, MediaPipe Pose로 머리·어깨 좌표를 추출해 분석. **자세 지표 계산** - EHD : 귀–어깨 간 수평 거리로 머리 전방 기울기 측정. - DPR : 얼굴 크기 변화를 이용해 거리 변화 판단. - 두 지표를 종합해 '**정상·주의·위험**' 상태로 분류. **로컬 데이터 저장 및 시각화** - 계산된 지표와 점수를 SQLite에 저장하고, IPC 통신으로 데이터를 전달해 차트로 표시. **시각적 피드백** - 상태별 색상으로 피드백 제공, 비정상 자세가 지속되면 알림 표시.

코드 주요 설명	(사진 1, 사진 2)카메라를 통해서 측정한 값(pitch, yaw, roll)을 내부에서 update 함수를 통해서 계산하여 결과값을 바탕으로 IPC channel 통신하여 사용자에게 전달하는 핵심로직을 반복하여 프로그램이 동작합니다.
개발 내용	**온디바이스 처리 구조 설계** - Electron 메인 프로세스와 렌더러 간 IPC 통신을 통해 AI 연산과 UI 처리를 분리. - 모든 MediaPipe 연산을 로컬에서 수행하여 프라이버시와 안정성을 확보했다. **성능 최적화** - MediaPipe Pose를 통한 사용자 움직임 추이를 이용한 CPU 연산 최적화 - 사용자 사용 환경에 맞게끔 측정 성능을 변경 할 수 있는 선택지 부여 **UI/UX 개선** - 자세 상태를 색상 및 그래프 형태로 표시하여 시각적 피드백을 강화. - Electron 창 크기와 위치를 기억해 재실행 시 사용자 맞춤 환경을 제공한다. 데이터 로깅 및 통계 기능 - SQLite에 저장된 자세 데이터를 기반으로 일·주 단위 통계 제공. - 사용자는 자신의 자세 개선 추이를 차트로 쉽게 확인할 수 있다.
시연 영상	
실행 파일	
기타 (선택)	민영재 : 앱 개발 담당 (화면 설계 및 구현, UI·UX 개선) 손준영 : AI 담당 (포즈 추정 모델 적용, 지표 계산 알고리즘 구현), 발표 최효원 : 앱 개발 담당 (화면 설계 및 구현, 성능 최적화, 패키징)

붙임1

```ts
update(input: MetricProcessorInput): MetricValues {
  const { landmarks, imageWidth, imageHeight, frameId, timestamp } = input;

  const face: FaceLandmarks | null | undefined = landmarks?.face ?? null;
  const pose: PoseLandmarks | null | undefined = landmarks?.pose ?? null;

  const resolvedTimestamp = resolveTimestamp(timestamp);
  const deltaSeconds = getDeltaSeconds(this.lastTimestamp, resolvedTimestamp);
  this.lastTimestamp = resolvedTimestamp;

  this.latencyStats?.record(deltaSeconds);

  const ehdResult = computeEhd(pose);
  const headPoseResult = computeHeadPose(face, imageWidth, imageHeight);
  const dprResult = computeDpr(face, this.baselineFaceSize);

  if (
    this.baselineFaceSize === null &&
    dprResult.confidence === "HIGH" &&
    Number.isFinite(dprResult.size ?? NaN)
  ) {
    this.baselineFaceSize = dprResult.size ?? null;
  }

  const reliability = landmarks?.reliability ?? "UNKNOWN";
  const frameConfidence = computeFrameConfidence(landmarks);

  this.signalProcessor.beginFrame(frameConfidence);
```

1) 사진 1

1) `apps/desktop/src/worker/metrics/index.ts` 내부 update 함수

```
this.signalProcessor.beginFrame(frameConfidence);

const processingContext: SignalProcessingContext = {
  deltaSeconds,
  reliability,
  frameConfidence,
};

const ehdSignal = this.signalProcessor.process(
  {
    key: "ehd",
    rawValue: ehdResult.value,
    confidence: ehdResult.confidence,
    source: ehdResult.source,
  },
  processingContext,
);

const pitchSignal = this.signalProcessor.process(
  {
    key: "pitch",
    rawValue: headPoseResult.pitch,
    confidence: headPoseResult.confidence,
    source: headPoseResult.source,
  },
  processingContext,
);

const yawSignal = this.signalProcessor.process(
  {
    key: "yaw",
    rawValue: headPoseResult.yaw,
    confidence: headPoseResult.confidence,
    source: headPoseResult.source,
  },
  processingContext,
);

const rollSignal = this.signalProcessor.process(
  {
    key: "roll",
    rawValue: headPoseResult.roll,
    confidence: headPoseResult.confidence,
    source: headPoseResult.source,
  },
  processingContext,
);
```

2)사진 2

2)`apps/desktop/src/worker/metrics/index.ts` 함수 내부