제주해양경찰서 **AI Agent** 시연 시스템 개발 일정표 **(1**인 개발자**)**

프로젝트 개요

- 개발 기간: 4주 (20일)
- 개발 인원: 1명
- 근무 시간: 주 5일 (월-금), 일 8시간 (오전 4시간, 오후 4시간)
- 목표: 핵심 기능이 구현된 시연 가능한 프로토타입 개발

개발 범위 **(**시연용 최소 기능**)**

1. 통합상황실 Agent (중앙 제어)
2. 경비구난 Agent (핵심 기능)
3. 민원상담 Agent (대민 서비스)
4. 웹 기반 관리자 대시보드
5. 모바일 반응형 사용자 인터페이스

제**1**주**:** 기반 구축 및 핵심 **Agent** 개발

Day 1 (월요일**)**

오전 **(09:00-13:00)**

- 09:00-10:00: 개발 환경 설정
  - AWS 계정 생성 및 기본 설정
  - DeepSeek API 키 발급 및 테스트
  - Git 저장소 생성 및 프로젝트 구조 설정
- 10:00-11:00: 개발 도구 설치 및 구성
  - Node.js, Python 환경 설정
  - Docker 설치 및 컨테이너 환경 구축
  - VS Code 및 필수 확장 프로그램 설치
- 11:00-13:00: 프로젝트 기본 구조 생성
  - Frontend (React) 프로젝트 초기화
  - Backend (FastAPI) 프로젝트 초기화
  - 모노레포 구조 설정 및 패키지 관리

오후 **(14:00-18:00)**

- 14:00-16:00: 시스템 아키텍처 설계
  - 전체 시스템 구조도 작성

    - API 엔드포인트 설계
    - 데이터베이스 스키마 설계 (PostgreSQL)
- 16:00-18:00: 기본 인프라 구축
    - Docker Compose 파일 작성
    - 로컬 개발 환경 자동화 스크립트 작성
    - 환경 변수 관리 시스템 구축

Day 2 (화요일**)**

오전 **(09:00-13:00)**

- 09:00-11:00: DeepSeek API 연동 모듈 개발
    - API 클라이언트 클래스 구현
    - 에러 처리 및 재시도 로직 구현
    - 응답 캐싱 시스템 구현
- 11:00-13:00: 기본 프롬프트 엔지니어링
    - 시스템 프롬프트 템플릿 작성
    - 역할별 프롬프트 구조 설계
    - 프롬프트 버전 관리 시스템 구축

오후 **(14:00-18:00)**

- 14:00-16:00: 데이터베이스 구축
    - PostgreSQL 컨테이너 설정
    - 기본 테이블 생성 (users, conversations, agents)
    - SQLAlchemy ORM 모델 정의
- 16:00-18:00: 인증 시스템 구현
    - JWT 기반 인증 구현
    - 사용자 등록/로그인 API 개발
    - 권한 관리 미들웨어 구현

Day 3 (수요일**)**

오전 **(09:00-13:00)**

- 09:00-11:00: 통합상황실 Agent 개발 시작
    - Agent 기본 클래스 구조 구현
    - 대화 컨텍스트 관리 시스템 개발
    - 메시지 라우팅 로직 구현
- 11:00-13:00: Agent 간 통신 시스템 구현
    - 메시지 큐 시스템 설계 (Redis 활용)
    - Agent 간 데이터 전달 프로토콜 정의
    - 비동기 통신 처리 구현

오후 **(14:00-18:00)**

- 14:00-16:00: 통합상황실 Agent 핵심 기능 구현
    - 사용자 의도 분석 기능
    - 적절한 하위 Agent 선택 로직

    - 응답 통합 및 포맷팅 기능
- 16:00-18:00: WebSocket 실시간 통신 구현
    - FastAPI WebSocket 엔드포인트 구현
    - 실시간 대화 스트리밍 기능
    - 연결 관리 및 에러 처리

Day 4 (목요일**)**

오전 **(09:00-13:00)**

- 09:00-11:00: 경비구난 Agent 개발
    - SAR (수색구조) 시나리오 프롬프트 작성
    - 위치 정보 처리 로직 구현
    - 긴급도 판단 알고리즘 개발
- 11:00-13:00: 해양 데이터 통합
    - 기상청 API 연동 (해양 기상 정보)
    - 좌표 계산 및 거리 측정 함수 구현
    - 모의 VTS 데이터 생성기 개발

오후 **(14:00-18:00)**

- 14:00-16:00: 경비구난 Agent 고급 기능
    - 수색 패턴 생성 알고리즘 구현
    - 자원 배치 최적화 로직 개발
    - 상황 보고서 자동 생성 기능
- 16:00-18:00: Agent 테스트 및 디버깅
    - 단위 테스트 작성 (pytest)
    - 통합 테스트 시나리오 작성
    - 성능 측정 및 최적화

Day 5 (금요일**)**

오전 **(09:00-13:00)**

- 09:00-11:00: 민원상담 Agent 개발
    - 자주 묻는 질문 데이터베이스 구축
    - 자연스러운 대화 처리 로직 구현
    - 다국어 지원 기반 구축 (한국어, 영어)
- 11:00-13:00: 민원 처리 워크플로우 구현
    - 민원 접수 및 분류 시스템
    - 담당 부서 자동 배정 로직
    - 처리 상태 추적 시스템

오후 **(14:00-18:00)**

- 14:00-16:00: 1주차 통합 테스트
    - 전체 Agent 간 연동 테스트
    - 엔드투엔드 시나리오 테스트
    - 버그 수정 및 안정화

- 16:00-18:00: 문서화 및 주간 정리
    - API 문서 자동 생성 (Swagger)
    - 개발 진행 상황 정리
    - 다음 주 계획 수립

제**2**주**: Frontend** 개발 및 시스템 통합

Day 6 (월요일**)**

오전 **(09:00-13:00)**

- 09:00-11:00: React 프로젝트 구조 설계
    - 컴포넌트 계층 구조 설계
    - 상태 관리 설정 (Redux Toolkit)
    - 라우팅 구조 설정 (React Router)
- 11:00-13:00: UI 프레임워크 설정
    - Tailwind CSS 설정 및 커스터마이징
    - 해양경찰 브랜드 컬러 팔레트 정의
    - 반응형 레이아웃 그리드 시스템 구축

오후 **(14:00-18:00)**

- 14:00-16:00: 인증 UI 개발
    - 로그인/로그아웃 컴포넌트 개발
    - 사용자 프로필 관리 UI
    - 권한별 접근 제어 구현
- 16:00-18:00: 메인 레이아웃 구현
    - 헤더/사이드바/푸터 컴포넌트
    - 네비게이션 메뉴 구현
    - 다크모드 지원 구현

Day 7 (화요일**)**

오전 **(09:00-13:00)**

- 09:00-11:00: 대화 인터페이스 개발
    - 채팅 UI 컴포넌트 개발
    - 메시지 버블 및 타임스탬프
    - 타이핑 인디케이터 구현
- 11:00-13:00: 실시간 통신 연동
    - WebSocket 클라이언트 구현
    - 메시지 스트리밍 처리
    - 재연결 로직 구현

오후 **(14:00-18:00)**

- 14:00-16:00: Agent 선택 인터페이스
    - Agent 카드 컴포넌트 개발
    - 드래그 앤 드롭 상호작용
    - Agent 상태 표시 (온라인/오프라인)
- 16:00-18:00: 대화 히스토리 관리
    - 대화 목록 컴포넌트
    - 검색 및 필터링 기능
    - 대화 내보내기 기능

Day 8 (수요일**)**

오전 **(09:00-13:00)**

- 09:00-11:00: 관리자 대시보드 개발
    - 시스템 상태 모니터링 위젯
    - 실시간 통계 차트 (Chart.js)
    - Agent 성능 메트릭 표시
- 11:00-13:00: 지도 기반 인터페이스
    - Leaflet 지도 통합
    - 제주 해역 커스텀 오버레이
    - 실시간 선박 위치 표시 (모의 데이터)

오후 **(14:00-18:00)**

- 14:00-16:00: 경비구난 전용 UI
    - SAR 작전 계획 도구
    - 수색 구역 그리기 도구
    - 자원 배치 시각화
- 16:00-18:00: 알림 시스템 구현
    - 브라우저 푸시 알림
    - 인앱 알림 센터
    - 알림 설정 관리

Day 9 (목요일**)**

오전 **(09:00-13:00)**

- 09:00-11:00: 모바일 반응형 최적화
    - 모바일 전용 네비게이션
    - 터치 제스처 지원
    - 모바일 성능 최적화
- 11:00-13:00: PWA 기능 구현
    - Service Worker 설정
    - 오프라인 모드 지원
    - 앱 설치 프롬프트

오후 **(14:00-18:00)**

- 14:00-16:00: 파일 업로드 기능

    - 드래그 앤 드롭 파일 업로드
    - 이미지 미리보기
    - 문서 파싱 및 분석
- 16:00-18:00: 다국어 지원 구현
    - i18n 설정
    - 한국어/영어 번역 파일
    - 언어 전환 UI

Day 10 (금요일**)**

오전 **(09:00-13:00)**

- 09:00-11:00: 통합 테스트 환경 구축
    - E2E 테스트 설정 (Cypress)
    - 주요 사용자 시나리오 테스트
    - 크로스 브라우저 테스트
- 11:00-13:00: 성능 최적화
    - 코드 스플리팅 적용
    - 이미지 최적화
    - API 요청 최적화

오후 **(14:00-18:00)**

- 14:00-16:00: 2주차 통합 점검
    - 전체 기능 연동 테스트
    - UI/UX 일관성 점검
    - 버그 수정
- 16:00-18:00: 배포 준비
    - 프로덕션 빌드 설정
    - 환경별 설정 분리
    - 배포 스크립트 작성

제**3**주**:** 고급 기능 구현 및 데이터 통합

Day 11 (월요일**)**

오전 **(09:00-13:00)**

- 09:00-11:00: 모의 데이터 생성 시스템
    - 실시간 선박 움직임 시뮬레이터
    - 기상 데이터 생성기
    - 사건/사고 시나리오 생성기
- 11:00-13:00: 데이터 스트리밍 파이프라인
    - Kafka 컨테이너 설정
    - 실시간 데이터 처리 로직
    - 데이터 변환 및 정규화

오후 **(14:00-18:00)**

- 14:00-16:00: 분석 대시보드 개발
    - 실시간 해상 교통 분석
    - 위험 구역 히트맵
    - 통계 보고서 생성
- 16:00-18:00: 예측 모델 통합
    - 간단한 ML 모델 구현 (사고 위험도)
    - 모델 서빙 API 구현
    - 예측 결과 시각화

Day 12 (화요일**)**

오전 **(09:00-13:00)**

- 09:00-11:00: Agent 학습 시스템
    - 대화 로그 수집 및 저장
    - 피드백 수집 메커니즘
    - 성능 메트릭 추적
- 11:00-13:00: A/B 테스트 프레임워크
    - 프롬프트 버전 관리
    - 실험 그룹 할당 로직
    - 결과 분석 도구

오후 **(14:00-18:00)**

- 14:00-16:00: 보안 강화
    - API Rate Limiting 구현
    - SQL Injection 방지
    - XSS 보호 적용
- 16:00-18:00: 감사 로그 시스템
    - 모든 API 호출 로깅
    - 사용자 활동 추적
    - 로그 검색 및 필터링

Day 13 (수요일**)**

오전 **(09:00-13:00)**

- 09:00-11:00: 백업 및 복구 시스템
    - 자동 백업 스케줄러
    - 데이터베이스 덤프 및 복원
    - 파일 시스템 백업
- 11:00-13:00: 모니터링 시스템 구축
    - Prometheus 메트릭 수집
    - Grafana 대시보드 설정
    - 알림 규칙 설정

오후 **(14:00-18:00)**

- 14:00-16:00: API 문서화 강화
    - OpenAPI 스펙 완성
    - 인터랙티브 API 문서
    - 사용 예제 추가
- 16:00-18:00: SDK 개발
    - Python SDK 기본 구조
    - JavaScript SDK 기본 구조
    - 사용 가이드 작성

Day 14 (목요일**)**

오전 **(09:00-13:00)**

- 09:00-11:00: 시연 시나리오 개발
    - 5개 주요 시나리오 스크립트 작성
    - 시연용 데이터 준비
    - 시나리오별 체크리스트
- 11:00-13:00: 시연 모드 구현
    - 데모 데이터 자동 생성
    - 시연 전용 UI 모드
    - 빠른 리셋 기능

오후 **(14:00-18:00)**

- 14:00-16:00: 부하 테스트
    - JMeter 테스트 시나리오 작성
    - 동시 사용자 100명 시뮬레이션
    - 병목 지점 분석
- 16:00-18:00: 성능 튜닝
    - 데이터베이스 인덱스 최적화
    - 캐싱 전략 개선
    - API 응답 시간 단축

Day 15 (금요일**)**

오전 **(09:00-13:00)**

- 09:00-11:00: 통합 리허설
    - 전체 시스템 동작 점검
    - 시연 시나리오 실행
    - 이슈 사항 기록
- 11:00-13:00: 긴급 수정
    - 발견된 버그 수정
    - UI 미세 조정
    - 성능 문제 해결

오후 **(14:00-18:00)**

- 14:00-16:00: 문서 최종 정리

    - 사용자 매뉴얼 완성
    - 관리자 가이드 완성
    - FAQ 문서 작성
- 16:00-18:00: 3주차 마무리
    - 코드 정리 및 리팩토링
    - 주석 및 문서화 보완
    - 최종 주 계획 수립

제**4**주**:** 최종 완성 및 시연 준비

Day 16 (월요일**)**

오전 **(09:00-13:00)**

- 09:00-11:00: AWS 배포 환경 구축
    - EC2 인스턴스 생성 및 설정
    - RDS PostgreSQL 설정
    - S3 버킷 구성
- 11:00-13:00: CI/CD 파이프라인 구축
    - GitHub Actions 워크플로우 작성
    - 자동 테스트 및 배포 설정
    - 환경별 배포 전략

오후 **(14:00-18:00)**

- 14:00-16:00: 프로덕션 배포
    - 애플리케이션 배포
    - 도메인 및 SSL 설정
    - CDN 구성 (CloudFront)
- 16:00-18:00: 배포 검증
    - 모든 기능 동작 확인
    - 성능 테스트
    - 보안 스캔

Day 17 (화요일**)**

오전 **(09:00-13:00)**

- 09:00-11:00: 시연 자료 준비
    - 프레젠테이션 슬라이드 작성
    - 시스템 아키텍처 다이어그램
    - 핵심 기능 스크린샷
- 11:00-13:00: 시연 동영상 제작
    - 주요 기능 데모 녹화
    - 음성 해설 추가
    - 편집 및 자막 추가

오후 **(14:00-18:00)**

- 14:00-16:00: 시연 환경 최적화
    - 시연 전용 데이터 세트 구성
    - 네트워크 안정성 확보
    - 백업 시연 환경 준비
- 16:00-18:00: 리허설 #1
    - 전체 시연 시나리오 실행
    - 시간 측정 및 조정
    - 예상 질문 리스트 작성

Day 18 (수요일**)**

오전 **(09:00-13:00)**

- 09:00-11:00: 사용자 피드백 수집
    - 베타 테스터 초대
    - 피드백 폼 구성
    - 실시간 피드백 세션
- 11:00-13:00: 최종 개선사항 적용
    - UI/UX 미세 조정
    - 발견된 버그 수정
    - 성능 최적화

오후 **(14:00-18:00)**

- 14:00-16:00: 보안 최종 점검
    - 취약점 스캔 도구 실행
    - 보안 체크리스트 검토
    - 민감 정보 노출 확인
- 16:00-18:00: 백업 및 복구 테스트
    - 전체 시스템 백업
    - 복구 절차 테스트
    - 재해 복구 계획 문서화

Day 19 (목요일**)**

오전 **(09:00-13:00)**

- 09:00-11:00: 최종 시연 리허설
    - 실제 환경에서 전체 시연
    - 발표자 스크립트 최종 검토
    - 기술 지원팀 역할 분담
- 11:00-13:00: 예상 Q&A 준비
    - 기술적 질문 답변 준비
    - 비즈니스 가치 설명 준비
    - 향후 계획 설명 자료

오후 **(14:00-18:00)**

- 14:00-16:00: 시연 자료 최종 점검
    - 모든 자료 최종 검토
    - 백업 자료 준비
    - 오프라인 시연 대비
- 16:00-18:00: 팀 브리핑
    - 시연 당일 계획 공유
    - 비상 대응 계획 수립
    - 역할 및 책임 최종 확인

Day 20 (금요일**)**

오전 **(09:00-13:00)**

- 09:00-10:00: 시연 환경 최종 점검
    - 모든 시스템 상태 확인
    - 네트워크 연결 테스트
    - 시연 장비 준비 상태 확인
- 10:00-11:00: 시연 실시
    - 오프닝 프레젠테이션 (10분)
    - 라이브 데모 (30분)
    - Q&A 세션 (20분)
- 11:00-13:00: 피드백 수집 및 정리
    - 참석자 피드백 수집
    - 개선 사항 기록
    - 향후 개발 방향 논의

오후 **(14:00-18:00)**

- 14:00-16:00: 프로젝트 문서 최종 정리
    - 기술 문서 완성
    - 소스 코드 정리
    - 프로젝트 인수인계 문서 작성
- 16:00-17:00: 프로젝트 회고
    - 성과 분석
    - 교훈 정리
    - 개선점 도출
- 17:00-18:00: 프로젝트 종료
    - 최종 백업
    - 리소스 정리
    - 프로젝트 공식 종료

주요 산출물

기술 산출물

1. 작동하는 프로토타입 시스템
    - 3개의 AI Agent (통합상황실, 경비구난, 민원상담)
    - 웹 기반 관리자 대시보드
    - 모바일 반응형 사용자 인터페이스
2. 소스 코드 및 문서
    - Git 저장소의 전체 소스 코드
    - API 문서 (Swagger UI)
    - 기술 아키텍처 문서
3. 배포 가능한 시스템
    - Docker 이미지
    - AWS 배포 스크립트
    - CI/CD 파이프라인

비즈니스 산출물

1. 시연 자료
    - 프레젠테이션 슬라이드
    - 데모 시나리오 스크립트
    - 시연 동영상
2. 문서
    - 사용자 매뉴얼
    - 관리자 가이드
    - 프로젝트 최종 보고서
3. 향후 계획
    - 전체 시스템 구현 로드맵
    - 예산 및 일정 계획
    - 확장 가능성 분석

리스크 관리

주요 리스크 및 대응 방안

1. 시간 부족
    - 핵심 기능에 집중
    - 일부 기능은 목업으로 대체
    - 야근 최소화를 위한 효율적 일정 관리
2. 기술적 난이도
    - 복잡한 기능은 단순화
    - 오픈소스 라이브러리 적극 활용

    - 필요시 외부 자문 요청
3. 시연 당일 장애

    - 로컬 백업 환경 준비
    - 녹화된 데모 영상 준비
    - 오프라인 프레젠테이션 자료 준비

이 일정표는 1명의 개발자가 4주 동안 시연 가능한 수준의 프로토타입을 개발하는 현실적인 계획입니다. 핵심 기능에 집중하고, 시연 효과를 극대화할 수 있는 기능들을 우선적으로 구현하도록 설계되었습니다.