

14일간의 계획: 일일 실행 가이드

디자인 확정 (1~2일차) → API 명세 정의 (3일차) → 백엔드/프론트엔드 병렬 개발 (4~11일차) → 통합 및 테스트 (12~14일차)

일차	날짜 (8월)	일일 목표	백엔드 (유지호, 서윤하)	프론트엔드 (임재호, 장병헌, 전연수)	디자인 (장병헌) / PM (임재호)	핵심 마일스톤 및 결과물
1	15 (금)	기반 구축 및 환경 설정	유지호: MVP 기능 명세 기반 DB 스키마(ERD) 설계. 서윤하: AWS 환경(RDS, EC2 등) 및 Git 레포지토리 설정.	임재호: React Native 프로젝트 초기화, 기본 네비게이션 구조(로그인, 회원가입, 대시보드) 설정. 전연수: 상태 관리 라이브러리(R edux Toolkit 등) 설정 및 학습.	장병헌: 모든 MVP 화면(로 그인, 회원가입 , 대시보드 , 퀘스트, 이벤트 추가)의 와이어프 레임 Figma에 서 확정. 임재호: 탐원 전체와 MVP 범위 최종 합의.	Git 레포지토리 생성. 프로젝트 초기 구조 완성.
2	16 (토)	디자인 확정 및 API 계약 초안	유지호: ERD 기반 DB 모델 구현. 사용자 인증 로직 개발 착수. 서윤하: 인증 로직 지원, CI/CD 파이프라인 구축 시작.	임재호/전연수 : 와이어프레임 기반으로 기능 없는 정적 UI 컴포넌트(버튼 , 입력창, 카드 등) 사전 제작.	장병헌: 로그인, 회원가입 , 대시보드 화면의 고품질(H igh-fidel ity) 디자인 Figma로 전달.	핵심 화면 Figma 디자인 전달 완료.

					임재호: API 엔드포인트 계약서 초안 작성.	
3	17 (일)	API 계약 확정 및 UI 골격 완성	유지호: API 엔드포인트 계약서(Swagger/P ostman) 최종 확정 및 공유. POST /api/auth/register 및 login 구현 완료.	임재호: 대시보드 화면 정적 UI 구현. 전연수: 퀘스트 목록 화면 정적 UI 구현. 장병현: '이벤트 추가' 모달/화면 정적 UI 구현.	장병현: 캐릭터 레벨별 이미지 등 모든 시각적 에셋(아 이콘, 이미지) 제공. 임재호: 유지호와 API 계약서 최종 검토 및 확정.	API 계약서 확정. 모든 정적 UI 화면 골격 완성.
4	18 (월)	사용자 인증 흐름 구현	유지호: GET /api/dashboard 엔드포인트 구현 (우선 고정된 Mock 데이터 반환). 서윤하: 배포 스크립트 작성.	임재호: register 및 login API 연동. JWT 토큰 저장 및 인증 기반 라우팅 구현. 전연수: 로그인/회원가 입 UI 마무리.	장병현: 프론트엔 드 팀의 UI 구현 지원 및 수정.	사용자 회원가입 및 로그인 기능 완료.
5	19 (화)	대시보드 데이터 연동	유지호: POST /api/chronicle/eve nts 엔드포인트 기본 로직 구현. (DB 저장까지만) 서윤하: API 테스트 케이스 작성 시작.	임재호: GET /api/dashboar d API 연동하여 대시보드에 사용자 이름, 크레도 점수(Mock	임재호: 팀 전체 진행 상황 점검 및 일정 조정.	대시보드에 서버 데이터(Mo ck) 표시 성공.

				데이터) 표시. 전연수: 캐릭터 이미지 교체 로직 프로토타입 구현.		
6	20 (수)	핵심 로직: 이벤트 생성 (1)	유지호: POST /api/chronicle/ev ents 호출 시, 규칙 기반으로 크레도 점수 재계산 및 DB 업데이트 로직 구현.	임재호: '이벤트 추가' 모달에서 입력받은 데이터를 POST /api/chronicle /events API로 전송하는 기능 구현.	장병현: 이벤트 카테고리 선택 UI 디자인 및 컴포넌트 구현 지원.	사용자가 입력한 이벤트가 DB에 저장됨.
7	21 (목)	핵심 로직: 이벤트 생성 (2)	유지호: 이벤트 생성 API 응답에 변경된 점수(new_cred o_score) 포함하여 반환. 서윤하: 이벤트 생성 API 테스트.	임재호/전연수 : 이벤트 생성 성공 후, 응답으로 받은 새 점수를 앱 상태(State)에 반영하고 대시보드 UI를 즉시 업데이트하는 로직 구현.	임재호: 중간 점검. 핵심 가치 루프의 절반 완성 점검.	이벤트 생성 → 점수 상승 → UI 반영 루프 완성.
8	22 (금)	퀘스트 기능 구현 (백엔드)	유지호: GET /api/quests (정적 퀘스트 목록 반환) 및 POST /api/quests/{id}/co mplete (퀘스트 완료 처리 및 점수 반영) 엔드포인트 구현.	전연수: GET /api/quests API 연동하여 퀘스트 목록 화면에 데이터 표시. 임재호: 대시보드 캐릭터 이미지 교체 로직 최종 구현.	장병현: 퀘스트 목록 UI 최종 검토 및 개선.	퀘스트 관련 백엔드 API 전체 구현 완료.
9	23 (토)	퀘스트 기능 구현 (프론트)	전연수: 퀘스트 '완료' 버튼 클릭 시 POST	임재호: 최근 이벤트 목록을 대시보드에	임재호: 최종 시연	퀘스트 완료 → 점수 상승

		엔드)	/api/quests/{id}/complete API 호출 및 UI 상태 업데이트 로직 구현.	표시하는 기능 구현. 장병헌: 전체적인 앱 UI/UX 통일성 검토 및 개선.	시나리오 초안 작성.	→ UI 반영 루프 완성.
10	24 (일)	1차 통합 및 버그 수정	유지호/서윤하: 프론트엔드 연동 과정에서 발생하는 API 관련 버그 수정 및 안정화 작업. 로깅(Logging) 강화.	전체: 개발된 모든 기능을 하나의 앱으로 통합하여 테스트. 회원가입부터 이벤트/퀘스트 완료까지 전체 플로우를 점검하며 발견되는 버그 수정.	장병헌: 최종 시연에 사용할 미래 기능 목업(Figma) 디자인 제작 시작.	주요 기능 통합 및 1차 버그 수정 완료.
11	25 (월)	디자인 개선 및 최종 테스트	유지호/서윤하: 서버 성능 모니터링 및 예외 처리 강화.	전체: UI/UX 최종 다듬기. 애니메이션 효과 추가 등 사용 경험 개선. 다양한 기기 환경에서 테스트.	장병헌: 미래 기능 목업 디자인(자동 피드, AI 추천, 금융 연계) 완성. 임재호: 최종 발표 자료 제작 시작.	기능 개발 동결 (Feature Freeze).
12	26 (화)	데모 준비 및 리허설	유지호/서윤하: 배포된 앱 모니터링. 심각한 버그 발생 시 긴급 대응.	전체: 최종 시연의 '골든 패스(Golden Path)' 스크립트에 맞춰 발표 리허설. 예상	임재호: 발표 자료 완성 및 팀원들과 최종	데모 시연 준비 및 리허설 완료.

				질문 및 답변 준비.	검토.	
13	27 (수)	최종 리허설 및 코드 동결	유지호/서윤하: 서버 안정성 최종 점검.	전체: 최종 리허설을 통해 시연 흐름을 완벽하게 숙지. 발견된 사소한 UI 버그 수정.	임재호: 발표 자료 및 시연 관련 모든 산출물 최종 정리.	코드 동결 (Code Freeze). 데모 완벽 숙지.
14	28 (목)	해커톤 준비	N/A (발표 지원)	N/A (발표 지원)	전체: 최종 발표 및 프로젝트 시연 준비.	프로젝트 제출 완료.