**2025 Us:Code 해커톤**

**팀 기술문서 양식(본선)**

| **팀명** | 세대초월 |
| --- | --- |
| **팀장** | 송은서 |
| **팀원 수 / 팀원 이름** | (총원 4명) / 오정민 이지혜 이가희 |
| **기술 스택 및 역할** | 1. **React(Nextjs)**   사용자 인터페이스(UI)를 만들기 위해 프론트 영역에서 활용하는 JavaScript 라이브러리. React는 동적인 모던 웹에서 빠른 퍼포먼스를 구현 가능함, 빠른 개발속도, 코드 유지보수 탁월성 등을 근거로 이를 적용함. 본 서비스 내에서 처음 시작 시 동적으로 빠르게 움직이는 실버가디언 로고가 바로 리액트를 활용한 부분이며, Nextjs를 통하여 RSC(React Server Component)를 위주로 사용하여 JavaScript 번들 사이즈를 줄이고 정적인 렌더링 위주로 일어나게 하여 저사양에서도 문제 없이 작동하는 웹 기반 UI 소프트웨어를 개발 및 배포함.   1. **Devup UI**   Rust기반의 CSS-IN-JS 스타일의 UI 라이브러리임. 저사양 모바일에서도 더욱 빠르게 렌더링이 될 필요가 있고 스타일 연산 최적화를 위하여 적용함. 해당 라이브러리가 적용된 제품의 성능 테스트의 결과는 아래와 같이 성능 분야에서 만점을 받음(100점 만점에 100점 달성). 그 외에도 “접근성”, “권장사항”, “검색엔진최적화” 등 대부분의 분야에서 굉장히 높은 점수를 달성함.     1. **Fastapi**   Python 기반의 Rest API를 구축하기 위한 고성능 프레임워크로 개발과 요청 처리 속도가 빠른 것이 장점이며, 데이터 검증 및 직렬화 작업을 Rust 기반의 Pydantic을 통해 자동 검증할 수 있는 것이 특징임. 본 서비스 내에서 User, Patrol 등 사용자의 정보를 이용한 CRUD API 구축에 적극 활용되었으며, 프론트엔드 직군과 더 빠른 소통을 하기 위하여 OPEN API를 기반으로 스키마를 공유하여 오차 없는 API를 기반으로 소통을 이어가며 개발을 진행할 수 있게 됨. 본 서비스의 경우 AI를 활용하는 것이 포함되어 있으므로 Python을 기반으로 서버 및 코어 로직을 구현함.   1. **Docker**   리눅스 컨테이너스기술로 만들어진 컨테이너 기술임. Docker로 코드 및 프로젝트를 패킹하여 서버에 올릴 수 있으며, 이는 도커 런타임만 있다면 어떤 호스트라도 같은 결과를 보장하므로, 유동성있는 서버를 구현하기 위하여 로드 밸런서와 함께 많이 사용되는 조합임. 본 프로젝트에서는 GCP CloudRun 인프라에 업로드하기 위하여 선택이 필요하였으며, 그 외에도 기본적인 프로젝트의 런타임 의존성 관리를 적절하게 관리하기 위하여 선택됨.   1. **GCP CloudRun**   서버리스(Serverless) 서비스로, 사용할 때만 서버를 활성화하기 때문에 가성비 및 병렬처리가 장점이며, 주로 Docker로 패킹하여 서버에 배포함. 따라서 앞선 Docker와 연계하여 가성비 좋은 서버리스 서비스를 손쉽게 배포할 수 있음. 서비스의 특정 시간에 병렬처리가 필요한 트래픽이 발생할 수 있으며, 날씨에 따라서 트래픽이 거의 발생하지 않는 경우가 있으므로 필요할 때만 병렬처리 및 버스팅이 필요한 상황에 맞게 인프라를 선택함.   1. **GCP SQL(Postgresql)**   GCP(Google Cloud Platform)에서 Postgresql 기반의 클라우드 데이터 베이스를 구축함. Postgresql는 복잡한 쿼리, 공간 데이터 등 다양한 유형을 지원할 수 있고, 관리형 데이터베이스를 사용함으로써 인프라/보안 측면에서 장점을 가짐. 본 서비스에서 사용자 정보를 저장하는데 사용함.   1. **MCP(Model Context Protocol)**   AI 모델이 다양한 도구나 데이터 소스에 접근할 수 있도록 하는 표준 프로토콜임. STDIO, HTTP, SSE 등 여러 전송 및 통신 방법을 제공하고 있으며, 대부분의 AI 가 MCP를 지원함. AI와 관계가 없는 서비스 또한 쉽게 MCP를 구축하여 AI와 연계할 수 있다는 장점을 지니고 있음. 본 서비스는 국가 공공 데이터에 접근하기 위하여 MCP 프로토콜을 활용했고, 이를 위하여 국가 공공 데이터에 대한 MCP를 직접 구축하였음. 이번에 구축한 MCP는 필요한 데이터를 실시간으로 끌어오는데, MCP는 이는 총 4가지 기능을 갖고 있음. MCP를 통하여 GEMINI와 통신을 하고 GEMINI는 해당 정보를 기반으로 산불의 발생 정도 등을 추측하므로 국가 기관 데이터의 MCP화는 이번 산불 예방을 궁극적 목표로 하는 본 서비스의 핵심적 가치에 부합하는 중요한 요소임을 알 수 있음.   1. **Tauri**   HTML, CSS, JS 프론트로 데스크톱 앱 및 모바일 앱을 만들게 해주는 프레임워크임. [Node.js](http://node.js) 와 다르게 Rust 를 사용하여 가볍고 빠르다는 장점이 존재하며 저사양 PC에서도 원활히 동작한다는 장점을 기반으로 선정함. 본 서비스는 스마트폰 및 피쳐폰 등 다양한 디바이스에서 지원이 가능해야한다는 점에 입각해, Tauri를 활용한 서비스를 제작하였으며, 기존 웹앱을 구현하는 기술을 그대로 유지하고 사용함에도, 이를 오프라인으로 이용할 수 있도록 전환이 가능하다는 매우 큰 장점을 갖고 있으므로, 오지에서 네트워크가 작동하지 않아 앱 자체가 먹통이 되는 문제를 해결하기 위하여 사용함.   1. **LangChain**   대규모 언어 모델을 활용한 애플리케이션 개발에 특화된 오픈소스 프레임워크임. 데이터베이스, API, 파일 시스템 등 과의 표준화된 인터페이스를 제공할 뿐 아니라 다양한 데이터를 실시간으로 활용할 수 있다는 점이 특징임. 따라서 빠르고 정확한 정보의 가공이 필요한 본 서비스에 부합한 프레임워크이므로 사용함.   1. **Github Action**   GitHub에서 공식적으로 제공하는 CI/CD 자동화 도구로 쉬운 유지보수와 로그 추적이 특징임. 본 서비스에서는 이 장점을 기반으로 GCP 배포 스크립트를 작성하여 CI/CD 환경을 구축하고, GCP CloudRun에 FastAPI 서버를 배포 함. 프론트엔드 빌드 및 배포 과정을 통합하는 워크플로우를 구축하는데 활용함. |
| **기술 제약 및 극복** | 1. **애플리케이션의 작동 환경에 따른 제약과 한계 극복**   서비스 구현 초기 기획으로 Flutter 혹은 ReactNative를 이용한 앱 내에 WebView 호출 및 구성을 통한 웹앱을 계획함. 하지만 본 서비스의 의의인 산불 예방 측면을 고려해 보았을 때, 해당 애플리케이션이 산 속에서 사용될 가능성이 매우 높으며, 산악 지형 특성상 인터넷이 원활하지 않을 수 있다는 문제점에 직면할 수 있음. 인터넷 환경에서 작동하는 인앱 브라우저 특성상, 인앱 브라우저를 반드시 사용할 수 밖에 없는 웹앱의 한계에 직면함. 토의 과정을 거쳐 구현 난이도가 어려워지는 위험을 무릅쓰고 Rust 기반의 크로스플랫폼 프레임워크인 Tauri 를 적극적으로 활용함. Tauri 는 네트워크가 없이도 내장된 경량 브라우저를 기반으로 웹앱을 빌드 시에 포함한 리소스를 이용하여 서비스를 전개하므로, 산 속과 같은 장애 환경 속에서도 본 서비스가 정상적으로 작동할 수 있도록 함.   1. **공공 데이터 기반 서비스와 데이터의 MCP 포팅 및 응용**   산림청에서 제공하는 공공 데이터의 갱신률이 저조하고, 군 이하로는 데이터의 수집률이 저조하여, 상세한 데이터를 기반으로 재난 예방을 목적으로하는 실버 가디언에 적용하기에 한계가 있음을 인지함. 따라서 작은 정보들이지만 흩어져 있는 디테일한 정보들을 MCP를 통하여 AI가 사용할 수 있는 형태로 재가공하였음. 결과적으로 기존의 산림청이 제공하는 데이터와 의성군의 온도, 습도, 산불 위험도 데이터를 AI가 즉각적으로 MCP를 통하여 사용할 수 있도록 설계하였고, 이를 통해 유기적인 종합 데이터를 추출하고 실시간으로 해당 읍의 유의미한 산불위험정도를 예측해 냄. 이는 나아가 MCP의 특성에 근거하여 구글의 다음 AI 모델의 등장에도 큰 제약없이 다음 세대의 AI를 수용할 수 있는 확장 플랫폼을 사용했다는 점에서 의의가 있음. |
| **기술 차별점** | 1. 앱 구현의 부가적 목적인 고령층 건강 증진 및 사회 복지 비용 절감을 실현하기 위해, 걸음 수 측정 프로그램 UI/UX를 도입함. 그러나 기존의 걸음 수 측정 앱과 차별성이 없다는 점을 발견했고, 성과 급여 시스템 연계를 설계 및 기획함.. 유저의 걸음 수 및 시간, GPS 정보를 이용해 유저에게 할당 된 순찰 경로를 완수 했는지 확인 후 걸음 수에 비례하여 성과를 급여하며 자의적 참여를 독려하는 시스템을 구성함. 이는 산불 예방에 이바지할 수 있을 뿐 아니라, 순찰 성과만큼 급여를 받을 수 있으며, 고령층의 경우 건강 복지 측면에 대한 비용이 절감되는 등의 기대효과를 지님.      1. 주 사용자들을 고령자들로 설정했을 때 고령자들을 위한 인터페이스를 기획하기위해 간단하고 객관적인 인터페이스를 제작함. 예를 들어 크기가 큰 글자를 배치하고 여러 네비게이션 바로 화면을 분류하지 않고 슬라이드 형식을 이용해 탭 수를 줄임. 그러나 기존의 고령자를 대상으로하는 어플과 차이점이 있다고 할 수 없기에, 회원가입 시스템에 변화를 줌. 앱을 처음 실행했을 때, 화면엔 유저의 uuid를 QR 코드로 구현하여 QR코드만을 보여줌. 이를 공무원에게 보여주면 공무원이 코드를 촬영함. 그러면 유저의 uuid가 관리자 데이터베이스에 입력되고 앱 내의 QR코드가 사라지고 앱이 실행됨. 추후 자신의 QR 코드는 설정에서 확인 가능함. 이를 통해 지자체에 순찰 인원으로써 개인정보를 넘겨줌과 동시에 회원가입 시스템을 대체할 수 있음. 결과적으로 디지털 포용성과 기술 접근성 향상 효과를 보임.      1. 기존의 재난 신고 앱들은 주로 사후 신고에 초점을 두고 있는 반면, 본 서비스인 실버가디언은 사전 예방부터 사후 대응까지 전과정을 아우르는 포괄적 시스템임. 또한 Gemini AI 등을 활용하여 산불 고위험 지역을 예측하여 순찰 인력을 배치하는 등 예방 측면에 힘쓰는 형태를 띄고 있음.      1. 의성군 특유의 지역 특성을 반영하여 이를 해결하고자 하는 서비스를 고안함. 그러나 의성군에 국한되는 문제가 아님을 제시하고 이를 점차 확장시켜 의성군과 유사한 케이스의 지역에 적용을 제시함. 더 나아가 범국가적인 시스템으로의 확장을 궁극적 목표로 하는 시스템을 고안함에 있어서 독창성이 존재함. |
| **기술 아키텍처** |  |
| **코드** | https://github.com/eunseo9311/UsCode\_SilverGuardian |

| **앱 배포 링크(IOS)** | <https://drive.google.com/file/d/1F75T-UtATUEcQBpoI7eZm7w2w9C2Xy5t/view?usp=sharing> |
| --- | --- |

| **앱 배포 링크(ANDROID)** | <https://drive.google.com/file/d/1FKA3nDtXJjZw1Zq19DXzDl246EG_Dz13/view?usp=sharing> |
| --- | --- |