

Chapter 1_시작하기 전에.md

시작하기 전에

가이드 목적

이 안내서는 SK hynix 내 비개발자 출신 엔지니어들이 사내 Cloud Platform인 HCP (Hybrid Cloud Platform)을 통해 웹 개발을 직접 진행할 수 있도록 가이드를 제공하는 목적으로 작성되었습니다. 기반기술센터 AI/DT TF 내 솔루션 분과를 담당하는 엔지니어들이 웹 기반 AFM Data Viewer Application을 만드는 과정에서 필요한 지식들을 담았습니다.

이 가이드는 AFM (Atomic Force Microscopy)라는 특정 장비를 대상으로 진행하였지만, FTP로 직접 데이터를 추출하거나 Data Lake를 통해 Smart Factory의 데이터를 얻을 수 있다면 반도체 제조 현장의 모든 장비에 적용할 수 있습니다.

대상 독자

이 가이드는 다음과 같은 분들을 대상으로 작성되었습니다.





주요 대상자

- 반도체 장비 관련 업무를 담당하는 비개발자 출신 엔지니어
- Python, JavaScript에 대한 기초적인 이해가 있거나 배울 의지가 있는 분
- 장비 데이터를 더 효과적으로 시각화하고 관리하고 싶은 분
- 기본적인 영어 읽기 능력이 있으신 분 (기술 문서 대부분이 영어로 작성됨)
- Code 작성 시 LLM을 활용해보신 분

웹 개발 기초 이해하기

웹 애플리케이션이란?

웹 애플리케이션은 인터넷 브라우저에서 실행되는 프로그램입니다. 여러분이 매일 사용하는:

-  네이버, 구글 (검색 사이트)
-  유튜브 (동영상 사이트)
-  온라인 쇼핑몰
-  우리가 만드는 AFM 데이터 뷰어

모두 웹 애플리케이션입니다!

웹의 3가지 핵심 기술

1. HTML (HyperText Markup Language)

역할: 웹페이지의 구조와 내용을 정의

```
<!-- 이런 식으로 생겼습니다 -->
<h1>제목입니다</h1>
<p>이것은 문단입니다</p>
<button>버튼입니다</button>
```

 **비유:** 집의 뼈대나 기둥 같은 역할


- 어디에 제목이 올지
- 어디에 버튼이 있을지
- 어디에 이미지가 들어갈지

2. CSS (Cascading Style Sheets)

역할: 웹페이지의 디자인과 레이아웃을 담당

```
/* 이런 식으로 생겼습니다 */
h1 {
  color: blue;
  font-size: 24px;
}

button {
  background-color: green;
  border-radius: 5px;
}
```

 **비유:** 집의 인테리어나 페인트 같은 역할

- 글자 색깔과 크기
- 버튼의 색깔과 모양
- 레이아웃 배치

3. JavaScript

역할: 웹페이지에 동작과 상호작용을 추가

```
// 이런 식으로 생겼습니다
function searchData() {
```

```

    console.log("검색 버튼을 클릭했습니다!");
}

const searchResults = [];

```

⚡ **비유:** 집의 전기, 수도, 엘리베이터 같은 역할

- 버튼을 클릭했을 때 무엇을 할지
- 데이터를 어떻게 처리할지
- 사용자와 어떻게 상호작용할지

웹 개발을 레스토랑으로 이해하기

웹 개발을 레스토랑 운영에 비유해서 이해해보겠습니다. 고객이 레스토랑에서 식사하는 전체 과정이 웹 애플리케이션의 동작 과정과 매우 유사합니다.

프론트엔드 (Frontend) - 고객이 직접 보고 경험하는 부분

프론트엔드는 레스토랑의 홀과 같습니다. 고객이 직접 보고 만지고 경험하는 모든 것들이 여기에 해당됩니다. 웹 브라우저에서 사용자가 보는 화면, 버튼을 클릭하는 상호작용, 아름다운 디자인과 애니메이션 등이 모두 프론트엔드의 영역입니다. HTML로 구조를 만들고, CSS로 예쁘게 꾸미며, JavaScript로 동적인 기능을 구현합니다.

백엔드 (Backend) - 고객이 보지 못하지만 핵심적인 처리가 일어나는 부분

백엔드는 레스토랑의 주방과 같습니다. 고객의 주문을 받아서 요리를 만들고, 재료를 관리하며, 결제를 처리하는 모든 뒷단 작업이 이루어지는 곳입니다. 웹에서는 데이터베이스에서 정보를 가져오고, 복잡한 계산을 수행하며, 보안을 관리하고, 다른 시스템과 연동하는 작업들이 백엔드에서 처리됩니다.

간단한 웹페이지 만들어보기

실제로 HTML, CSS, JavaScript가 어떻게 작동하는지 간단한 예제로 확인해보겠습니다. 확장자 .html 파일을 만들고 아래 내용을 입력해 봅니다.

```

<!DOCTYPE html>
<html>
  <head>
    <title>내 첫 웹페이지</title>
    <style>
      body {
        font-family: Arial, sans-serif;
        text-align: center;
        padding: 50px;
      }

      .welcome-box {

```

```
        background-color: #f0f8ff;
        border: 2px solid #4caf50;
        border-radius: 10px;
        padding: 20px;
        margin: 20px auto;
        max-width: 400px;
    }

    button {
        background-color: #4caf50;
        color: white;
        padding: 10px 20px;
        border: none;
        border-radius: 5px;
        cursor: pointer;
        font-size: 16px;
    }

    button:hover {
        background-color: #45a049;
    }
</style>
</head>
<body>
    <div class="welcome-box">
        <h1>안녕하세요! 🙌</h1>
        <p>웹 개발의 세계에 오신 것을 환영합니다!</p>
        <p id="message">버튼을 클릭해보세요</p>
        <button onclick="showMessage()">클릭하세요!</button>
    </div>

    <script>
        function showMessage() {
            document.getElementById("message").innerHTML =
                "🎉 축하합니다! JavaScript가 실행되었습니다!";

            // 버튼 텍스트도 변경
            event.target.innerHTML = "다시 클릭!";
        }
    </script>
</body>
</html>
```

이 예제에서 배울 수 있는 것들

- **HTML:** 구조를 만들고 내용을 정의 (<h1> , <p> , <button>)
- **CSS:** 예쁜 디자인과 색상, 레이아웃 적용
- **JavaScript:** 버튼 클릭 시 동적으로 내용 변경

프로젝트 기술 구성

이번 프로젝트에서 사용한 기술들을 앞서 설명한 프론트엔드와 백엔드로 구분해서 살펴보겠습니다. HCP에서 제공하는 환경을 최대한 활용하여 안정적이고 확장 가능한 시스템을 구축하는 것이 목표입니다.

백엔드 기술 스택

Python 3.11

Python은 배우기 쉬우면서도 강력한 프로그래밍 언어입니다. 특히 데이터 분석과 과학 계산 분야에서 널리 사용되기 때문에, 반도체 장비 데이터를 처리하는 우리의 목적에 부합합니다. HCP Web App에서는 다양한 Python 버전 (3.8~3.11)을 제공하며 우리는 3.11을 선택하였습니다. 벤치마크 상, Python 3.11은 이전 버전들과 비교해 약 10-60%의 성능 향상을 보여주며, 특히 수치 계산과 텍스트 처리에서 성능 개선을 보입니다. 또한 더 명확한 에러 메시지를 제공하여 디버깅이 훨씬 쉬워졌습니다. (Python 공식 사이트)

Flask 웹 프레임워크

Flask는 Python으로 웹 애플리케이션을 만들 수 있게 해주는 마이크로 프레임워크입니다. 복잡하지 않으면서도 필요한 기능들을 충분히 제공하기 때문에, 웹 개발이 처음인 엔지니어들에게 적합합니다. 마치 레고 블록처럼 필요한 기능들을 하나씩 조립해가며 웹 서버를 만들 수 있습니다. Django와 같은 풀스택 프레임워크와 달리 가볍고 유연하여, 우리가 필요한 기능만 선택적으로 추가할 수 있습니다. (Flask 공식 문서)

프론트엔드 기술 스택

웹 애플리케이션을 위해서는 사용자가 직접 상호작용하는 프론트엔드 기술들이 필요합니다. 이번 프로젝트에서는 현대적이면서도 비교적 학습하기 쉬운 기술들을 선택했습니다.






(예시) 프로젝트 구조 한눈에 보기

```
front-end/
├── public/           # 정적 파일들
│   └── favicon.ico  # 브라우저 탭 아이콘
├── src/              # 소스 코드 (우리가 주로 작업하는 곳)
│   ├── assets/      # 이미지, 로고 등 리소스
│   ├── components/  # 재사용 가능한 UI 조각들
│   ├── layouts/     # 페이지 레이아웃 (헤더, 푸터)
│   ├── pages/       # 각 페이지별 화면
│   ├── router/      # 페이지 간 이동 설정
│   ├── stores/      # 데이터 저장소
│   ├── styles/      # 스타일 설정
│   ├── dummy/       # 테스트용 가짜 데이터
│   └── main.js       # 앱 시작점
└── docs/            # 개발 가이드 문서
```

```
├── package.json
└── README.md
```

```
# 프로젝트 설정과 의존성
# 프로젝트 설명서
```

각 폴더의 역할

-  **pages/**: 사용자가 보는 각 화면 (레고의 완성품)
-  **components/**: 재사용 가능한 UI 조각들 (레고 블록)
-  **layouts/**: 모든 페이지에 공통으로 들어가는 부분 (건물의 로비)
-  **stores/**: 여러 페이지가 공유하는 데이터 저장소 (창고)
-  **assets/**: 이미지, 아이콘 등 (장식품)

HTML/CSS/JavaScript

웹 페이지의 기본 구성 요소들입니다. HTML은 내용의 구조를 만들고, CSS는 시각적인 디자인을 담당하며, JavaScript는 사용자와의 상호작용을 처리합니다. 이 세 가지 기술은 모든 웹 개발의 기초가 되며, 브라우저에서 직접 실행되어 사용자에게 즉각적인 반응을 제공합니다. 우리 애플리케이션은 Chrome 브라우저에서 최적화되어 동작하도록 개발됩니다.

Node.js

Node.js는 JavaScript를 브라우저 밖에서도 실행할 수 있게 해주는 런타임 환경입니다. 전통적으로 JavaScript는 웹 브라우저 안에서만 작동했지만, Node.js 덕분에 서버 개발, 개발 도구 제작, 심지어 데스크톱 애플리케이션까지 만들 수 있게 되었습니다. 우리 프로젝트에서는 프론트엔드 개발 도구들을 실행하고, 패키지를 관리하며, 빌드 과정을 자동화하는 데 사용됩니다. 마치 Python이 데이터 분석뿐만 아니라 웹 개발에도 사용되는 것처럼, JavaScript도 Node.js를 통해 다양한 영역에서 활용할 수 있습니다. (Node.js 공식 사이트)

Vue.js

Vue.js는 사용자 인터페이스를 구축하기 위한 JavaScript 프레임워크입니다. 널리 알려진 React 프레임워크와 비교했을 때, Vue.js는 학습 곡선이 완만하고 직관적인 문법을 제공합니다. 특히 HTML과 CSS에 익숙한 개발자라면 쉽게 적응할 수 있으며, 점진적으로 도입할 수 있어 기존 프로젝트에 부분적으로 적용하기도 좋습니다. 또한 Vue.js는 한국어 문서가 잘 번역되어 있고, 아시아권에서 많이 사용되고 있습니다. (Vue.js 공식 사이트)

Vuetify

Vuetify는 Vue.js를 위한 UI 컴포넌트 라이브러리로, Google의 Material Design 기반으로 디자인된 아름다운 컴포넌트들(버튼, 카드, 테이블, 차트 등)을 제공합니다. 따라서 디자인에 대한 고민 없이 전문적인 외관의 웹 애플리케이션을 만들 수 있습니다. Vuetify 외에도 UI와 디자인은 개발자 취향에 맞게 다양하게 선택될 수 있습니다. 현대의 UI Component들은 반응형 디자인이 기본으로 적용되어 있어 데스크톱, 태블릿, 모바일에서 모두 잘 작동합니다. (Vuetify 공식 사이트) (Vue UI Component Library 리스트)

ECharts

ECharts는 Apache 소프트웨어 재단에서 관리하는 강력한 데이터 시각화 라이브러리입니다. 다양한 종류의 차트와 그래프를 지원하며, 인터랙티브한 기능들을 제공합니다. 반도체 장비에서 나오는 복잡한 데이터를 직관적인 차트로 변환하여 엔지니어들이 쉽게 이해할 수 있도록 도와줍니다. 선 그래프, 막대 차트, 히트맵, 3D 차트 등 다양한 시각화 옵션을 제공하며, 실시간 데이터 업데이트도 지원합니다. (ECharts 공식 사이트)

주요 용어 정리

프로젝트를 이해하는 데 도움이 되는 중요한 용어들입니다:

용어	의미	예시
Component	재사용 가능한 UI 조각	버튼, 카드, 헤더
Page	사용자가 방문하는 완전한 화면	홈페이지, 결과 페이지
Router	URL과 페이지를 연결하는 시스템	/ → 홈페이지
Store	여러 컴포넌트가 공유하는 데이터 저장소	로그인 정보, 검색 기록
API	서로 다른 프로그램이 소통하는 방법	프론트엔드 ↔ 백엔드
SPA	페이지를 새로고침하지 않는 웹앱	Vue.js로 만든 우리 앱

SK hynix HCP (Hybrid Cloud Platform) 활용

우리 회사의 클라우드 플랫폼인 HCP를 활용하여 다음과 같은 서비스들을 사용합니다. 비개발자 출신의 엔지니어들은 HCP와 같은 관리형 서비스들을 활용함으로써 인프라 구축과 관리에 드는 시간과 비용을 절약하고, 개발 자체에 집중할 수 있습니다.

Web App 서비스

HCP에서 제공하는 웹 애플리케이션 호스팅 서비스로, 우리가 개발한 Python Flask 애플리케이션을 배포하고 운영할 수 있습니다. 별도의 서버 구축 작업 없이 웹 애플리케이션 개발을 바로 진행할 수 있습니다.

VS Code 개발 환경

클라우드 기반의 통합 개발 환경으로, 웹 브라우저로 코딩을 할 수 있습니다. 로컬 컴퓨터에 개발 환경을 구축할 필요 없이, 공유된 환경에서 개발을 진행할 수 있어 팀 협업에 매우 유용합니다. 또한 모든 설정과 패키지가 클라우드에 저장되어 일관된 개발 경험을 제공합니다.

학습 목표

기술적 역량

웹 개발의 기본 개념을 이해하고, Python과 JavaScript를 활용하여 실제 동작하는 웹 애플리케이션을 개발할 수 있게 됩니다. 또한 데이터베이스와 연동하여 장비 데이터를 저장하고 조회하는 시스템을 구축할 수 있습니다. 더 나아가 Vue.js를 활용한 반응형 웹 애플리케이션 개발 능력도 갖추게 됩니다.

실무 적용 능력

단순한 예제가 아닌, 실제 업무에서 바로 활용할 수 있는 AFM 데이터 뷰어를 완성하게 됩니다. 이 경험을 바탕으로 다른 장비나 시스템에도 유사한 솔루션을 개발할 수 있는 자신감을 얻게 됩니다.

문제 해결 사고력

웹 개발 과정에서 발생하는 다양한 문제들을 스스로 해결하는 방법을 익히게 됩니다. 에러 메시지를 읽고 분석하는 방법, 공식 문서를 참조하는 방법 등을 배우게 됩니다.

사전 준비사항

웹 개발이 처음이라고 해서 걱정하실 필요는 없습니다. 이 프로젝트에 참여한 AI/DT 솔루션 분과 엔지니어들도 데이터 분석 경험은 있지만 웹 개발은 해본 적이 없는 분들이 대부분이었습니다. 따라서 이 가이드는 기본적인 프로그래밍 이해가 있는 분들이라면 차근차근 따라할 수 있도록 구성되어 있습니다.

각 단계마다 LLM을 통해 빠르게 코드를 생성할 수도 있지만, 처음 개발을 해보시는 분들은 코드를 직접 작성해보시고, 에러가 발생하더라도 포기하지 마시기 바랍니다. 에러는 학습의 일부이며, 문제를 해결해나가는 과정에서 더 깊은 이해를 얻을 수 있습니다. 완벽을 추구하기보다는 꾸준히 실습하면서 점진적으로 이해도를 높여가는 것이 중요합니다.

웹 애플리케이션의 장점

웹으로 개발된 애플리케이션은 업무 효율성을 크게 향상시킬 수 있습니다:

장점	효과	업무 적용
쉬운 접근	웹 브라우저만 있으면 OK	사내 인터넷 망으로 쉽게 데이터 확인
실시간 협업	여러 사람이 동시 작업	팀원 간 실시간 데이터 공유
통합 관리	데이터 일원화	Excel 파일 버전 관리 문제 해결
자동 업데이트	최신 정보 유지	수동 파일 동기화 불필요
확장성	사용자/기능 쉽게 추가	새로운 장비나 분석 기능 확장

AFM 데이터 관리의 변화

- Before: Eng'r가 매뉴얼로 데이터 확인 및 관리 → 업무 부담, 협업 어려움

- **After:** 웹 기반 통합 관리 → 실시간 공유, 자동 분석

실습해보기

1. **웹 브라우저 열기:** 크롬, 엣지 중 하나 선택
2. **개발자 도구 열기:** F12 키를 누르거나 우클릭 → "검사"
3. **간단한 코드 테스트:** Console 탭에서 다음 코드 입력해보기

```
// 브라우저 콘솔에서 실행해보세요
console.log("안녕하세요, 웹 개발 세계!");
alert("첫 번째 JavaScript 실행!");
```

다음 단계로

이제 기본 개념을 이해했으니, 다음 장에서 실제 개발 환경을 구축해보겠습니다!