# LostArk 예측 시스템 - Flask 웹 화면 구조 설계

## 1. 개요

본 문서는 Flask 기반으로 구성될 로스트아크 아이템 시세 예측 시스템의 웹 화면 구조를 정의합니다. 템플릿 구조, 정적 파일 디렉토리 구성, 주요 라우팅 흐름, API 연동 방식 등을 포함하여 향후 유지보수 및 확장을 용이하게 합니다.

## 2. 템플릿 구조 (templates/)

Flask에서 사용하는 Jinja2 템플릿 파일은 아래와 같이 구성됩니다:

* base.html - 공통 레이아웃, 헤더/푸터/JS 포함
* search.html - 아이템 검색 및 다중 선택 화면
* chart.html - 선택된 아이템의 시세 + 예측 차트 출력 화면

## 3. 정적 파일 구조 (static/)

정적 리소스는 다음과 같이 정리합니다:

* js/chart.js - 차트 렌더링 및 사용자 인터랙션 처리
* css/style.css - 전체 스타일 시트

## 4. Flask 라우트 설계

주요 라우팅은 아래와 같이 설계됩니다.

* GET / - search.html 렌더링 - 아이템 검색 및 선택
* POST /chart - 선택된 아이템, 예측 기간을 기반으로 chart.html 렌더링

## 5. FastAPI 연동 방식

Flask는 백엔드 예측 로직과 분리되어 있으며, FastAPI와의 연동을 통해 데이터를 받아옵니다.

예측 데이터는 JavaScript (fetch 또는 axios)를 통해 `/predict/{item\_id}` 등의 엔드포인트를 비동기 호출하여 가져옵니다.

## 6. 사용자 동작 흐름 요약

1. 1. 사용자 검색창에 아이템 이름 입력
2. 2. 여러 개의 아이템 선택
3. 3. 예측 기간(1/3/7일 등) 선택
4. 4. '보기' 버튼 클릭 시 /chart 로 POST 요청
5. 5. 차트 페이지(chart.html)에서 FastAPI 예측 데이터 fetch → 그래프 렌더링