# <UI/UX 정의서>

| UI/UX 정의서 | |
| --- | --- |
| 작성자 | 김동훈 |
| 작성 일자 | 2024/10/14 |
| 팀명 | Sniffers |

| 순서 | 작업 내용 | 페이지 |
| --- | --- | --- |
| 1 | 프로젝트 개요 | 3 |
| 프로젝트 이름 |
| 프로젝트 목적 |
| 주요 목표 |
| 2 | 사용자 페르소나 | 4 |
| 3 | 기능 요구 사항 | 5 |
| 데이터 업로드 | 5 ~ 6 |
| 이상 거래 탐지 결과 대시보드 | 7 ~ 8 |
| 이메일 알림 | 9 ~ 10 |
| 4 | 정보 구조 (Information Architecture) | 11 |
| 홈 (Home) | 11 ~ 12 |
| 데이터 업로드 (Upload Data) | 13 ~ 14 |
| 대시보드 (Dashboard) | 15 ~ 16 |
| 로그인/회원가입 페이지 | 17 ~ 18 |
| 5 | 사용자 인터페이스 디자인 (UI Design) | 19 |
| HTML 파일 설명 | 19 ~ 35 |
| CSS 파일 분석 | 36 ~ 41 |
| 6 | 시각적 요소 | 42 |
| 와이어프레임 및 시각적 디자인 | 42 ~ 44 |
| 색상 팔레트 및 폰트 | 45 ~ 46 |
| 7 | 사용자 경험 흐름 (UX Flow) | 47 |
| 로그인, 데이터 업로드, 대시보드 시각화 과정 | 47 ~ 49 |
| 이메일 알림 기능 | 50 ~ 52 |
| 8 | 접근성 고려사항 | 53 |
| 색상 대비 | 53 ~ 54 |
| 텍스트 크기 조정 가능 여부 | 55 ~ 57 |
| 스크린 리더 호환성 | 58 ~ 61 |

# <프로젝트 개요>

* 프로젝트 이름: 이상 거래 탐지 시스템 (Fraud Detection System)
* 프로젝트 목적:

이상 거래 탐지 시스템 (Fraud Detection System)의 목적은 사용자에게 트랜잭션 데이터를 업로드할 수 있는 직관적인 인터페이스를 제공하고, 업로드된 데이터를 기반으로 이상 거래 탐지 모델이 거래의 이상 여부를 판단하는 시스템을 제공하는 것이다. 탐지된 결과는 실시간으로 대시보드에서 시각화되며, 이상 거래 발생 시 사용하는 이메일로 알림을 받는다.

* 주요 목표:

1. 데이터 업로드 기능: 사용자가 손쉽게 트랜잭션 데이터를 CSV 형식으로 업로드할 수 있는 인터페이스를 제공.
2. 사기 탐지: 업로드된 데이터를 기반으로 사기 탐지 모델이 자동으로 거래의 이상 여부를 분석.
3. 시각화 대시보드: 탐지 결과를 실시간으로 대시보드에서 시각화하여 사용자에게 직관적인 정보 제공.
4. 이메일 알림 기능: 이상 거래 발생 시 즉시 사용자에게 이메일로 알림 발송.

# <사용자 페르소나>

**Primary Persona**

* 직업: 금융 데이터 분석가
* 목표: 트랜잭션 데이터를 통해 빠르고 정확하게 이상 거래를 탐지하고, 결과를 바탕으로 금융 리스크를 최소화하는 것이 목표
* 기술 수준: 중급 이상의 컴퓨터 및 데이터 분석 기술 보유
* 사용 동기: 수천 건의 거래 데이터를 처리하며, 수동 분석의 비효율성을 해소하고, 이상 거래 탐지 모델을 통해 자동으로 이상 거래를 탐지하기 위함.
* 특징:
  + 대량의 데이터를 효율적으로 처리하는 시스템이 필요
  + 실시간으로 거래 결과를 확인할 수 있는 시각화 도구가 필수적

**Secondary Persona**

* 직업: IT 관리자
* 목표: 사기 탐지 시스템을 운영하고, 데이터 보안을 유지하며, 서버를 관리하는 것이 목표
* 기술 수준: 고급 IT 지식과 서버 관리 경험이 풍부
* 사용 동기: 시스템이 안정적으로 운영되도록 하며, 데이터 업로드와 시각화 프로세스가 원활하게 작동하는지 모니터링하기 위함
* 특징:
  + 서버의 성능과 보안 상태를 지속적으로 점검해야 함
  + 시스템의 장애 발생 시 신속한 대처가 중요

# <기능 요구 사항>

### <데이터 업로드>

* 목적: 사용자가 CSV 파일을 업로드하여 이상 거래 탐지 분석을 수행할 수 있는 기능을 제공한다.
  + 설명:
    - 사용자는 CSV 파일 형식으로 트랜잭션 데이터를 시스템에 업로드할 수 있다.
    - 업로드된 데이터는 시스템 내에서 이상 거래 탐지 모델에 의해 분석된다.
  + 필수 데이터 컬럼:
    - Amount: 거래 금액
    - Customer ID: 고객 ID
    - Merchant ID: 상점 ID
    - Lat: 거래 위치의 위도
    - Long: 거래 위치의 경도
  + 업로드 방식:
    - 사용자는 업로드 폼에서 파일을 선택하여 업로드한다. (파일 형식은 CSV)
    - 업로드된 데이터는 Django 서버에서 처리되어 MySQL 데이터베이스에 저장된다.
  + 업로드 후 처리:
    - 업로드된 데이터는 사기 탐지 모델에 입력되어 분석이 진행된다.
    - 분석 결과는 Elasticsearch에 인덱싱 되어 Kibana에서 시각화된다.
  + 예외 처리:
    - 필수 컬럼이 누락된 경우, 오류 메시지를 사용자에게 표시하고 업로드를 중단한다.
    - CSV 파일의 형식이 잘못된 경우 업로드가 차단되며, 사용자는 오류 안내 메시지를 받는다.

### <이상거래 탐지 결과 대시보드>

* 목적: 사용자가 업로드한 데이터를 분석한 결과를 시각적으로 제공하여 이상 거래와 정상 거래를 쉽게 구분할 수 있도록 한다.
  + 설명:
    - 업로드된 트랜잭션 데이터를 바탕으로 사기 탐지 모델이 각 거래의 이상 거래 여부를 예측한다.
    - 예측 결과는 Kibana 대시보드에서 실시간으로 시각화된다.
    - 사용자는 대시보드를 통해 특정 고객 ID나 상점 ID별로 결과를 필터링하거나, 거래의 위치 정보를 지도에서 확인할 수 있다.
  + 시각화 요소:
    - 지도 시각화: 거래 위치(위도 및 경도)를 바탕으로 지도에서 거래 분포를 시각화한다.
      * 정상 거래는 파란색, 사기 거래는 빨간색으로 표시되어 한눈에 이상 거래를 파악할 수 있다.
    - 거래 목록 테이블: 거래 금액, 고객 ID, 상점 ID, 위치 정보 등을 포함한 거래 목록이 테이블 형식으로 제공된다.
    - 필터 기능: 사용자는 특정 거래 기간, 거래 금액 또는 고객 및 상점 ID를 기준으로 거래를 필터링할 수 있다.
  + 대시보드 구성:
    - 대시보드는 Elasticsearch에 저장된 데이터를 기반으로 Kibana에서 장동 생성된다.
    - 사용자별 대시보드 URL이 제공되며, 각 사용자는 본인의 데이터만을 조회할 수 있다.
  + 대시보드 업데이트:
    - 트랜잭션 데이터를 새로 업로드하면 대시보드가 자동으로 갱신된다.
    - 사용자 인터페이스에서 자동 새로고침 기능을 통해 실시간 데이터를 반영한다.
  + 예외처리:
    - 데이터가 없거나 이상 거래가 발견되지 않은 경우, 사용자는 “이상 거래 없음”이라는 메시지를 확인하게 된다.
    - Elasticsearch 또는 Kibana와의 연결이 끊긴 경우, 오류 메시지를 제공하고 사용자에게 대시보드가 로딩되지 않았음을 알린다.

### <이메일 알림>

* 목적: 이상 거래가 탐지될 경우 사용자에게 자동으로 이메일을 발송하여 경고 메시지를 전달한다.
  + 설명:
    - 사용자가 업로드한 데이터에서 이상 거래가 탐지되면, 시스템은 자동으로 사용자에게 이메일을 발송한다.
    - 이메일은 탐지된 이상 거래의 세부 정보를 포함하며, 이를 통해 사용자는 즉각적인 조치를 취할 수 있다.
* 이메일 발송 조건:
  + 탐지된 거래 중 이상 거래가 존재할 경우에만 이메일이 발송된다.
  + 이상 거래가 없으면 이메일이 발송되지 않는다.
* 이메일 내용:
  + 제목: [Sniffers] Fraudulent Transaction Detected!
  + 본문:
    - 거래 내역: 이상 거래로 탐지된 각 거래의 금액, 고객 ID, 상점 ID, 위치 정보(위도, 경도)가 포함된다.
    - 사용자는 이상 거래 발생 시간과 거래 세부 정보를 이메일에서 직접 확인할 수 있다.
* 이메일 템플릿:
  + HTML 템플릿: ‘fraud/email\_template.html에서 이메일 본문이 구성된다.
  + 내용 포맷: 이상 거래의 상세 정보를 표로 정리하여, 깔끔하게 정보를 제공한다.
* 이메일 발송 시스템:
  + SMTP 서버를 이용하여 이메일을 발송한다. 시스템 설정에서 SMTP 서버 정보가 미리 설정되어 있어야 한다.
  + 발송된 이메일은 사용자의 등록된 이메일 주소로 전달된다.
* 예외 처리:
  + 이메일 발송 중 문제가 발생하면 시스템은 오류 로그를 기록하고, 사용자에게 알림을 발송하지 못했음을 관리자에게 통보한다.
  + 사용자의 이메일 주소가 유효하지 않거나 이메일 서버와의 연결이 끊긴 경우, 시스템에서 실패 로그를 남긴다.

# <정보 구조>

### <홈 (Home)>

* 목적: 홈 페이지는 Fraud Detection System의 개요를 설명하고, 사용자에게 주요 기능에 대한 접근 경로를 제공하는 중앙 허브 역할을 한다.
  + 설명:
    - 사용자는 홈페이지에서 Fraud Detection System의 주요 기능 및 시스템 작동 방식을 소개받는다.
    - 홈 페이지는 네비게이션 메뉴를 통해 사용자에게 각 기능으로의 빠른 접근을 제공한다.
  + 주요 구성 요소:

1. 시스템 소개: 시스템이 어떻게 작동하는지에 대한 간단한 설명과 시스템의 주요 목적을 소개한다.
2. 네비게이션 바:

* 홈: 사용자가 현재 위치하고 있는 페이지.
* 대시보드: 사용자가 자신의 데이터를 기반으로 실시간 대시보드에서 이상 거래를 확인할 수 있는 링크
* 데이터 업로드: CSV 파일을 업로드하여 이상 거래 분석을 요청할 수 있는 업로드 페이지로 링크
* 로그인/로그아웃: 로그인 여부에 따라 로그인 또는 로그아웃 옵션이 제공된다.
* 회원가입: 신규 사용자가 시스템에 등록할 수 있는 링크

1. “About Us” 섹션: 시스템의 개발 목적, 기술 스택, 주요 기능 등을 간략히 소개하는 섹션.
2. “How It Works” 섹션: 사용자가 시스템을 어떻게 사용할 수 있는지 절차를 설명한다. 데이터 업로드부터 이상 거래 탐지까지의 과정이 요약된다.

* 네비게이션 경로:
  + /(Home): 홈페이지로 이동
  + /dashboard: 이상 거래 탐지 결과를 확인 할 수 있는 대시보드로 이동
  + /upload\_form: 데이터를 업로드하는 페이지로 이동
  + /login: 사용자가 시스템에 로그인할 수 있는 페이지로 이동
  + /singup: 사용자가 시스템에 회원가입할 수 있는 페이지로 이동

### <데이터 업로드 (Upload Data>

* 목적: 사용자가 이상 거래 탐지 분석을 위해 트랜잭션 데이터를 CSV 파일 형식으로 업로드할 수 있도록 지원하는 페이지이다.
  + 설명:
    - 사용자는 데이터 업로드 페이지에서 CSV 파일을 업로드하여, 시스템에 이상 거래 탐지 분석을 요청할 수 있다.
    - 페이지 상단에는 파일 업로드 규칙과 필수 컬럼에 대한 안내가 제공된다.
* 주요 구성 요소:

1. 업로드 안내 메시지: CSV 파일이 특정 형식을 준수해야 한다는 안내 메시지 (필수 컬럼: ‘Amount’, ‘Customer ID’, ‘Merchant ID’, ‘Lat’, ‘Long’)
2. 파일 선택 및 업로드 버튼: 사용자는 파일 선택을 통해 CSV 파일을 업로드할 수 있으며, 업로드 버튼을 눌러 데이터를 서버로 전송한다.
3. 에러 메시지 출력: CSV 파일에 문제가 있을 경우 (필수 컬럼 누락 등), 페이지에 오류 메시지가 빨간색으로 출력된다.

* 네비게이션 경로:
  + /upload\_form: 사용자가 CSV 파일을 업로드할 수 있는 페이지로 이동
  + 파일 업로드가 성공적으로 완료되면, 사용자는 자동으로 대시보드 페이지로 리다이렉트된다.
* 입력 및 출력:
  + 입력: 사용자가 업로드하는 CSV 파일
  + 출력: 업로드가 완료되면, 데이터는 이상 거래 탐지 분석에 사용되며, 결과는 대시보드에서 확인할 수 있다.
* 예외 처리:
  + 잘못된 형식의 파일을 업로드할 경우 오류 메시지가 표시된다.
  + 업로드된 파일이 CSV 형식이 아니거나 데이터가 올바르지 않으면 업로드가 차단된다.

<대시보드 (Dashboard)>

* 목적: 사용자가 업로드한 데이터를 바탕으로 이상 거래 탐지 결과를 시각화하여 제공하는 실시간 대시보드이다.
  + 설명:
    - 사용자가 업로드한 트랜잭션 데이터에 대한 분석 결과를 시각적으로 표시하는 페이지이다.
    - 대시보드는 지도와 거래 목록 테이블로 구성되어, 사용자는 거래의 위치 및 주요 정보를 한눈에 확인할 수 있다.
* 주요 구성 요소:

1. 지도 시각화:

* 위도(Lat)와 경도(Long)를 바탕으로 모든 거래의 위치가 지도에 표시된다.
* 정상 거래는 파란색, 이상 거래는 빨간색으로 표시되어 사용자가 시각적으로 즉각적인 구분이 가능하다.

1. 거래 목록 테이블:

* 거래의 세부 정보를 테이블 형식으로 표시하며, 거래 금액, 고객 ID, 상점 ID, 거래 위치, 이상 거래 여부 등의 정보가 포함된다.

1. 필터 및 정렬 기능:

* 사용자는 거래 금액, 고객 ID, 상점 ID, 등을 기준으로 거래를 필터링하거나 정렬할 수 있다.

1. 실시간 갱신:

* 대시보드는 실시간으로 갱신되어 새로운 데이터가 반영되며, 사용자는 최신 결과를 항상 확인할 수 있다.
* 네비게이션 경로
  + /dashboard: 사용자가 이상 거래 탐지 결과를 시각적으로 확인할 수 있는 대시보드 페이지
  + /dashboard2: 지도와 그래프를 이용해 사기 거래를 시각화하는 확장 대시보드 페이지 (해당 기능은 확장 가능성에 따라 사용)
* 입력 및 출력:
  + 입력: 시스템에서 이상 거래 탐지 모델을 통해 분석된 거래 데이터
  + 출력: 이상 거래 여부, 거래 위치, 거래 금액 등이 포함된 지도 시각화 및 거래 목록 테이블
* 예외 처리:
  + 거래 데이터가 없거나 이상 거래가 탐지되지 않은 경우, 사용자에게 “이상 거래 없음”이라는 메시지를 표시한다.
  + 대시보드 로딩 실패 시, 오류 메시지를 통해 사용자가 이를 인지할 수 있도록 한다.

### <로그인/회원가입 페이지>

* 목적: 사용자가 시스템에 로그인하거나 새 계정을 생성할 수 있도록 지원하는 페이지인다.
  + 설명:
    - 로그인 페이지에서는 이미 계정이 있는 사용자가 시스템에 접근할 수 있도록 로그인 폼을 제공한다.
    - 회원가입 페이지는 신규 사용자가 시스템에 가입할 수 있는 페이지로, 사용자명, 이메일, 비밀번호를 입력하여 계정을 생성할 수 있다.
* 주요 구성 요소:

1. 로그인 페이지 (login.html):

* 폼 요소: 사용자명과 비밀번호를 입력할 수 있는 필드 제공
* 로그인 버튼: 입력된 사용자명과 비밀번호를 바탕으로 인증 절차를 진행
* 오류 메시지 출력: 잘못된 사용자명 또는 비밀번호 입력 시 오류 메시지 표시
* 회원가입 링크: 신규 사용자는 회원가입 페이지로 이동할 수 있는 링크 제공

1. 회원가입 페이지 (signup.html):

* 폼 요소: 사용자명, 이메일, 비밀번호, 비밀번호 확인 필드를 포함
* 회원가입 버튼: 입력된 정보로 계정을 생성
* 오류 메시지 출력: 필수 필드 누락 또는 비밀번호 불일치 시 오류 메시지 출력
* 비밀번호 요구사항 안내: 비밀번호는 최소 8자 이상, 영문자와 숫자, 특수문자를 포함해야 한다는 안내 제공
* 네비게이션 경로:
  + /login: 사용자가 로그인할 수 있는 페이지로 이동
  + /signup: 신규 사용자가 회원가입할 수 있는 페이지로 이동
  + 로그인 성공 시 홈 페이지 또는 대시보드 페이지로 리다이렉트된다.
* 입력 및 출력:
  + 입력: 로그인 시 사용자명과 비밀번호, 회원가입 시 사용자명, 이메일, 비밀번호
  + 출력: 로그인 또는 회원가입 성공 시 홈페이지 또는 대시보드로 리다이렉트
* 예외 처리:
  + 로그인 정보가 잘못되었거나 회원가입 시 필수 정보가 누락된 경우, 오류 메시지가 표시된다.
  + 비밀번호 요구사항을 충족하지 못하면, 비밀번호 입력란 아래에 오류 메시지를 출력한다.

# <HTML 파일 설명>

### <upload\_form.html>

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Upload Form</title>

<link rel="stylesheet" href="{% static 'css/style.css' %}">

</head>

<body>

<header>

{% block content %}

<h1>Upload Transaction Data</h1>

<nav>

<ul>

<li><a href="{% url 'index' %}">Home</a></li>

<li><a href="{% url 'upload\_form' %}">Upload Data</a></li>

<li><a href="http://211.188.51.137:5601/app/dashboards#/view/58a42c20-7a19-11ef-9b66-13538d49ff90?\_g=(filters:!(),refreshInterval:(pause:!f,value:10000),time:(from:'1969-12-31T23:59:59.880Z',to:'1969-12-31T23:59:59.890Z'))">View Dashboard in Kibana</a></li>

<li><a href="{% url 'dashboard' %}">Dashboard</a></li>

</ul>

</nav>

{% endblock %}

</header>

<form method="POST" enctype="multipart/form-data">

{% csrf\_token %}

<p style="color:red">The CSV file has to include <b><i>'Amount', 'Customer ID', 'Merchant ID', 'Lat', and 'Long'</i></b> columns</p>

<label for="file">Choose a CSV file to upload:</label>

<input type="file" id="file" name="file" accept=".csv" required>

<button type="submit">Upload</button>

</form>

{% if error %}

<p style="color: red;">{{ error }}</p>

{% endif %}

</body>

</html>

| 기능 | |
| --- | --- |
| 사용자에게 거래 데이터를 업로드할 수 있는 폼을 제공하는 페이지이다. 사용자들은 CSV 파일을 업로드하여 거래 데이터를 시스템에 입력할 수 있으며, 필요한 CSV 파일의 컬럼들에 대한 지침도 제공한다. | |
| 코드 설명 | |
| <link rel = “stylesheet” href = “{% static ‘css/style.css’ %}”> | 정적 파일(css)을 로드하는 코드로, 스타일링을 적용하기 위해 사용된다. static 태그를 통해 Django의 정적 파일 관리 시스템에서 CSS 파일을 불러온다. |
| <form method = “POST” enctype = “multipart/form-data”>  …  <input type = “file” id = “file” name = “file” accept = “.csv” required>  …  </form> | 사용자에게 CSV 파일을 업로드할 수 있는 폼을 제공한다.  사용자가 CSV 파일을 선택하여 업로드할 수 있도록 설정한다. |
| {% csrf\_token %} | 보안을 위해 CSRF 토큰을 추가한다. |
| {% if error %}  …  {% endif %} | 업로드 과정 중 에러 발생 시 에러를 출력한다. |
| 입력 | |
| 사용자가 선택한 CSV 파일을 업로드한다. | |
| 출력 | |
| 서버에 제출된 CSV 파일이 처리되고, 업로드 성공 또는 오류 메시지가 출력된다. | |
| 매개변수 | |
| 파일 | CSV 파일 형식으로 업로드해야 하며, 특정 컬럼들이 반드시 포함되어야 한다. |
| 예외 처리 | |
| 파일 형식이나 업로드 오류가 있을 경우, 오류 메시지가 화면에 출력된다. | |
| 알고리즘 | |
| 폼을 통해 사용자가 파일을 업로드한 후, 서버에서 해당 파일을 처리하여 데이터베이스에 저장되도록 한다. | |
| 추가 설명 | |
| 이 페이지는 사용자들이 거래 데이터를 손쉽게 업로드할 수 있도록 하고, 해당 데이터는 분석 및 시각화를 위해 시스템에서 처리된다. 또한, 사용자들에게 CSV 파일에 반드시 포함되어야 할 컬럼들에 대한 명확한 지침을 제공한다. | |

### <dashboard.html>

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Fraud Detection Dashboard</title>

<link rel="stylesheet" href="{% static 'css/style.css' %}">

</head>

<body>

<header>

<h1>Fraud Detection Dashboard</h1>

<nav>

<ul>

<li><a href="{% url 'index' %}">Home</a></li>

<li><a href="{% url 'upload\_form' %}">Upload More Data</a></li>

<li><a href="http://211.188.51.137:5601/app/dashboards#/view/58a42c20-7a19-11ef-9b66-13538d49ff90?\_g=(filters:!(),refreshInterval:(pause:!f,value:10000),time:(from:'1969-12-31T23:59:59.880Z',to:'1969-12-31T23:59:59.890Z'))">View Dashboard in Kibana</a></li>

</ul>

</nav>

</header>

<iframe src="{{ kibana\_url }}" style="width:100%; height:100vh;" frameborder="0"></iframe>

</body>

</html>

| 기능 | |
| --- | --- |
| 사기 탐지 대시보드를 표시하는 페이지로, 사용자에게 Kibana 대시보드를 통해 실시간으로 사기 탐지 데이터를 시각화한다. 페이지 내에서 iframe을 사용하여 Kibana 대시보드를 임베드하여 보여준다. | |
| 코드 설명 | |
| <iframe src = “{{ kibana\_url }}” style = “width:100%, height:100vh;” frameborder = “0”></iframe> | Kibana 대시보드를 iframe으로 임베드하여 실시간 시각화를 제공한다.  {{ kibana\_url }}은 Django에서 전달된 Kibana 대시보드의 URL이다.  iframe은 전체 페이지에 걸쳐 시각화된 데이터를 표시하며, 대시보드의 시각적 업데이트가 실시간으로 반영된다. |
| 입력 | |
| 서버에서 제공되는 ‘kibana\_url’ 값이 iframe을 통해 사용된다. | |
| 출력 | |
| Kibana 대시보드가 실시간으로 임베드되어 보여진다. | |
| 매개변수 | |
| kibana\_url | 임베드된 Kibana 대시보드를 가리키는 URL |
| 예외 처리 | |
| ‘kibana\_url’ 값이 없거나 유효하지 않은 경우, 대시보드가 올바르게 표시되지 않을 수 있다. | |
| 알고리즘 | |
| 서버에서 전달된 Kibana 대시보드 URL을 iframe에 전달하여 실시간 데이터 시각화를 제공한다. | |
| 추가 설명 | |
| 이 페이지는 데이터 분석 및 사기 탐지 결과를 시각화하는 대시보드로, 사용자는 실시간으로 거래 데이터를 검토하고 분석할 수 있다. Kibana를 통해 제공되는 그래프와 차트는 자동으로 새로고침되어 최신 상태의 데이터를 반영한다. | |

### <login.html>

{% extends 'base.html' %}

{% load static %}

<head>

<link rel="stylesheet" href="{% static 'css/style.css' %}">

</head>

{% block content %}

<h2>Login</h2>

<form method="POST">

{% csrf\_token %}

<label for="username">Username:</label>

<input type="text" name="username" id="username">

<label for="password">Password:</label>

<input type="password" name="password" id="password">

<button type="submit">Login</button>

</form>

<p>Don't have an account? <a href="{% url 'signup' %}">Sign up here</a>.</p>

{% endblock %}

| 기능 | |
| --- | --- |
| 이 템플릿은 사용자 로그인을 위한 페이지를 제공한다. 사용자는 사용자명과 비밀번호를 입력하여 시스템에 로그인할 수 있으며, 아직 계정이 없는 사용자는 회원가입 링크를 통해 가입할 수 있다. | |
| 코드 설명 | |
| {% extends ‘base.html’ %} | 이 템플릿은 ‘base.html’ 파일을 확장하여 공통 레이아웃을 사용한다. 기본적인 페이지 구조는 ‘base.html’에서 정의되고, 현재 템플릿의 내용은 해당 구조 내에서 렌더링된다. |
| <form method = “POST”>  {% csrf\_token %}  …  </form> | 사용자가 로그인할 수 있는 폼을 제공한다.  보안을 위해 CSRF 토큰을 추가하여 폼 제출을 보호한다. |
| <a href = “{% url ‘signup’ %}></a> | 회원가입이 필요한 사용자를 위해 회원가입 페이지로 연결하는 링크를 제공한다. 링크는 Django의 URL 태그를 사용하여 동적으로 처리된다. |
| 입력 | |
| 사용자는 사용자명과 비밀번호를 입력하여 로그인 폼을 제출한다. | |
| 출력 | |
| 로그인 요청이 처리되며, 성공 시 사용자는 대시보드나 메인 페이지로 리다이렉트된다. 로그인 실패 시 오류 메시지가 표시될 수 있다. | |
| 매개변수 | |
| 사용자명 | 사용자가 입력하는 텍스트 필드 |
| 비밀번호 | 사용자가 입력하는 비밀번호 필드 |
| 예외 처리 | |
| 잘못된 사용자명이나 비밀번호가 입력될 경우, 서버에서 인증 오류가 발생하여 적절한 메시지를 반환할 수 있다. | |
| 알고리즘 | |
| 로그인 폼이 제출되면 서버에서 사용자의 자격 증명을 검증하고, 성공 시 사용자 세션을 생성하여 로그인 상태를 유지한다. | |
| 추가 설명 | |
| 이 페이지는 시스템에 로그인하기 위한 필수적인 부분이며, 사용자 인증 절차의 핵심 요소이다. 성공적인 로그인 후 사용자에게 적절한 권한을 부여하고, 로그인을 유지할 수 있도록 세션 처리가 이루어진다. | |

### <signup.html>

{% extends 'base.html' %}

{% load static %}

<head>

<link rel="stylesheet" href="{% static 'css/style.css' %}">

</head>

{% block content %}

<h2>Sign Up</h2>

<div style="color:white;">

<p><strong>Username Requirements:</strong></p>

<ul>

<li>The username is required.</li>

</ul>

<p><strong>Password Requirements:</strong></p>

<ul>

<li>Password must be at least 8 characters long.</li>

<li>Password must contain both letters and numbers.</li>

<li>Password must include at least one special character (e.g., !@#$%^&\*).</li>

<li>Password must not be a commonly used password (e.g., "password123", "qwerty", etc.).</li>

<li>Password cannot consist entirely of numbers (e.g., "12345678" is not allowed).</li>

</ul>

</div>

<form method="POST" class="signup-form-bottom-space">

{% csrf\_token %}

<label for="username">Username:</label>

<input type="text" name="username" id="username" value="{{ form.username.value }}">

<label for="email">Email:</label>

<input type="email" name="email" id="email" value="{{ form.email.value }}">

<label for="password1">Password:</label>

<input type="password" name="password1" id="password1">

<label for="password2">Confirm Password:</label>

<input type="password" name="password2" id="password2">

<button type="submit">Sign Up</button>

</form>

{% if form.errors %}

<div style="color:red;">

<ul>

{% for field in form %}

{% if field.errors %}

<li>{{ field.label }}: {{ field.errors|striptags }}</li>

{% endif %}

{% endfor %}

{% for error in form.non\_field\_errors %}

<li>{{ error|striptags }}</li>

{% endfor %}

</ul>

</div>

{% endif %}

{% endblock %}

| 기능 | |
| --- | --- |
| 사용자 회원가입을 위한 페이지로, 사용자가 필요한 정보를 입력하고, 제출 시 서버에서 처리하여 새로운 계정을 생성한다. 비밀번호 조건 및 필수 입력 사항에 대한 안내도 제공한다. | |
| 코드 설명 | |
| {% extends ‘base.html’ %} | 이 템플릿은 ‘base.html’을 확장하여 공통 레이아웃을 사용한다. 페이지의 기본 구조는 ‘base.html’에서 정의되고, 본문 내용만이 현재 템플릿에서 변경된다. |
| {% csrf\_token %} | 보안을 위해 CSRF 토큰을 추가하여 폼 제출을 보호한다. |
| {% if form.errors %}  …  {% endif %} | ‘form.errors’는 Django의 Form 객체에서 발생한 모든 에러를 포함하는 속성이다. 이 코드는 제출된 폼에 오류가 있는지 확인하는 조건문이다. 만약 ‘form.errors’가 존재하는 경우 (True) 블록 안의 내용을 실행한다. |
| {% for field in form %}  …  {% endfor %} | 폼의 각 필드를 순환하는 for 루프이다. ‘form’ 객체에는 폼에서 정의된 여러 필드들이 포함되어 있다. 이 루프를 통해 각 필드를 하나씩 확인하면서 필드에 에러가 있는지 검사한다. |
| {% if field.errors %}  …  {% endif %} | 개별 필드의 ‘errors’ 속성을 확인한다. 이 속성은 해당 필드에 대한 에러 메시지를 포함한다. 필드에 오류가 있으면(True), 내부의 블록이 실행된다. |
| <li>{{ field.label }}: {{ field.errors|striptags }}</li> | 필드의 레이블(예: Username, Password’)과 해당 필드의 에러 메시지를 출력하는 리스트 항목이다.  ‘{{ field.label }}’은 필드의 사용자 친화적인 이름을 출력한다. 예를 들어, ‘Username’ 또는 ‘Password’와 같은 값을 제공한다.  ‘{{ field.errors|striptags }}’는 해당 필드의 에러 메시지를 출력하는 부분이다. ‘|striptags’ 필터는 HTML 태그를 제거하여, 오류 메시지가 HTML로 렌더링되지 않도록 보장한다. 예를 들어 ‘<’와 같은 HTML 태그 문자가 포함된 오류 메시지를 깨끗한 텍스트로 표시한다. |
| {% for error in form.non\_field\_errors %}  …  {% endfor %} | 필드와 직접 연겨로디지 않은 비필드 오류를 처리하는 for 루프이다. 이 오류들은 폼 레벨에서 발생하는 오류로, 예를 들어 두 비밀번호 필드의 불일치와 같은 경우가 있다.  ‘form.non\_field\_errors’는 필드와 무관한, 폼 자체에서 발생하는 오류들을 포함한다. |
| <li>{{ error|striptags }}</li> | 비필드 오류 메시지를 리스트 항목으로 출력한다. ‘error’ 변수는 비필드 오류 메시지를 포함하며, ‘striptags’ 필터를 사용하여 HTML 태그를 제거하고 안전하게 출력한다. |
| 입력 | |
| 사용자는 사용자명, 이메일, 비밀번호, 비밀번호 확인 정보를 입력한다. | |
| 출력 | |
| 회원가입 요청이 처리되고, 성공 시 사용자 계정이 생성된다. 오류가 발생할 경우 적절한 오류 메시지가 출력된다. | |
| 매개변수 | |
| username, email, password1, password2 | 각 입력 필드의 값이 전달된다. |
| 예외 처리 | |
| 비밀번호 일치 여부, 형식 오류, 필수 값 누락 등의 상황에서 오류 메시지가 출력된다. | |
| 알고리즘 | |
| 사용자가 입력한 정보를 검증한 후, 서버에서 회원가입 처리를 수행한다. 모든 조건이 충족되면 사용자 계정이 생성된다. | |
| 추가 설명 | |
| 이 템플릿은 사용자 회원가입 절차를 지원하며, 계정 생성 시 필수 정보를 입력하도록 유도한다. 입력 점증 및 오류 처리가 포함되어 있어 사용자에게 올바른 정보를 제공할 수 있도록 설계되었다. | |

### <base.html>

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>{% block title %}My Website(DH){% endblock %}</title>

<link rel="stylesheet" href="{% static 'css/style.css' %}">

</head>

<body>

<header>

<h1>Welcome to the Fraud Detection System</h1>

<nav>

<ul>

<li><a href="{% url 'home' %}">Home</a></li>

<li><a href="{% url 'dashboard' %}">Dashboard</a></li>

<li><a href="{% url 'upload\_form' %}">Upload Data</a></li>

{% if user.is\_authenticated %}

<li><a href="{% url 'logout' %}">Logout</a></li>

{% else %}

<li><a href="{% url 'login' %}">Login</a></li>

<li><a href="{% url 'signup' %}">Sign Up</a></li>

{% endif %}

</ul>

</nav>

</header>

<main>

{% block content %}

{% endblock %}

</main>

<footer>

<p>© 2024 Fraud Detection System. All rights reserved.</p>

</footer>

</body>

</html>

| 기능 | |
| --- | --- |
| Django 애플리케이션의 공통 레이아웃을 정의하는 파일로, 다른 템플릿들이 이를 확장하여 공통 요소를 재사용할 수 있도록 한다. 제목, 헤더, 네비게이션 메뉴, 푸터와 같은 공통 구조를 제공하며, 페이지별 콘텐츠는 ‘{% block content %}’ 안에 렌더링된다. | |
| 코드 설명 | |
| {% load static %} | Django의 정적 파일을 로드하는 태그이다. ‘static’ 태그를 사용하여 CSS, JavaScript, 이미지 등 정적 파일을 사용할 수 있다. |
| <title>{% block title %}My Website(DH){% endblock %}</title> | HTML 페이지의 ‘<title>’ 태그를 정의하는 부분으로, 각 페이지가 이 템플릿을 확장하여 자신만의 제목을 설정할 수 있도록 한다. 기본 제목은 “My Website(DH)”이다.  ‘{% block title %}’은 확장된 템플릿에서 이 블록을 채워 페이지마다 다른 제목을 설정할 수 있다. |
| {% if user.is\_authenticated %}  …  {% else %}  …  {% endif %} | 현재 사용자가 로그인한 상태인지 확인한다.  로그인 상태일 경우 ‘Logout’ 링크가 표시되고, 비로그인 상태일 경우 ‘Login’과 ‘Sign Up’ 링크가 표시된다. |
| {% block content %}  …  {% endblock %} | 이 부분은 하위 템플릿에서 확장될 영역으로, 각 페이지마다 고유한 내용을 이 블록에 채울 수 있다. 이는 동적으로 페이지별 콘텐츠를 변경할 수 있도록 지원한다. |
| 입력 | |
| 이 템플릿은 다른 템플릿에서 ‘extends’를 사용하여 확장된다. 각 페이지의 고유한 콘텐츠는 ‘{% block content %}’ 안에 정의된다. | |
| 출력 | |
| 공통 레이아웃을 포함한 HTML 페이지가 생성된다. | |
| 매개변수 | |
| {% block title %}  {% block content %} | 각 페이지가 이 템플릿을 확장할 때 채울 수 있는 영역이다. |
| 예외 처리 | |
| 사용자가 로그인하지 않았을 경우 로그아웃 링크 대신 로그인 및 회원가입 링크가 표시된다. | |
| 알고리즘 | |
| 이 템플릿은 다른 페이지들이 기본적으로 사용할 공통 구조를 정의하고, 각 페이지의 콘텐츠와 제목은 하위 템플릿에서 동적으로 설정한다. | |
| 추가 설명 | |
| 이 템플릿은 프로젝트의 전체적인 레이아웃을 정의하며, 공통으로 사용되는 네비게이션과 푸터를 제공한다. 각 페이지는 이 템플릿을 확장하여 자신의 고유한 콘텐츠와 제목을 제공할 수 있다. | |

### <index.html>

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Fraud Detection System</title>

<link rel="stylesheet" href="{% static 'css/style.css' %}">

</head>

<body>

<header>

<h1>Welc<a href="https://vidkidz.tistory.com/192" class="easter-egg">o</a>me t<a href="https://vidkidz.tistory.com/85" class="easter-egg">o</a> the Fraud Detection System</h1>

<nav>

<ul>

<li><a href="{% url 'upload\_form' %}">Upload Data</a></li>

<li><a href="http://211.188.51.137:5601/app/dashboards#/view/58a42c20-7a19-11ef-9b66-13538d49ff90?\_g=(filters:!(),refreshInterval:(pause:!f,value:10000),time:(from:'1969-12-31T23:59:59.880Z',to:'1969-12-31T23:59:59.890Z'))">View Dashboard in Kibana</a></li>

<li><a href="{% url 'dashboard' %}">Dashboard (Overview)</a></li>

{% if user.is\_authenticated %}

<li><a href="{% url 'logout' %}">Logout</a></li>

{% else %}

<li><a href="{% url 'login' %}">Login</a></li>

<li><a href="{% url 'signup' %}">Sign Up</a></li>

{% endif %}

</ul>

</nav>

</header>

<main>

<h2>About Us</h2>

<p>This system helps in detecting fraudulent transacti<a href="https://vidkidz.tistory.com/29" class="easter-egg">o</a>ns effectively.</p>

<h2>How It Works</h2>

<p>Our model analyzes transaction patterns to identify potential fraud.</p>

</main>

<footer>

<p>&copy; 2024 Fraud Detection System. All rights reserved.</p>

</footer>

</body>

</html>

| 기능 | |
| --- | --- |
| Fraud Detection System의 메인 페이지를 정의한다. 사용자에게 데이터 업로드, 대시보드 보기, 로그인 및 로그아웃 링크를 제공하며, 시스템의 작동 방식에 대한 간단한 설명을 포함하고 있다. | |
| 코드 설명 | |
| {% load static %} | Django의 정적 파일을 로드하는 태그이다. ‘static’ 태그를 사용하여 CSS, JavaScript, 이미지 등 정적 파일을 사용할 수 있다. |
| {% if user.is\_authenticated %}  …  {% else %}  …  {% endif %} | 로그인 상태인지 확인하여 로그아웃 또는 로그인/회원가입 링크를 표시한다. |
| 입력 | |
| 사용자가 로그인 상태에 따라 다르게 표시되는 링크를 선택할 수 있다. | |
| 출력 | |
| Fraud Detection System의 메인 페이지가 렌더링된다. | |
| 매개변수 | |
| ‘{% block content %}’ 없음. 이 페이지는 고정된 내용을 포함하며, 확장되지 않는다. | |
| 예외 처리 | |
| 사용자가 로그인하지 않았을 경우, 로그아웃 링크 대신 로그인 및 회원가입 링크가 표시된다. | |
| 알고리즘 | |
| 로그인 상태와 관계없이 기본적인 네비게이션 메뉴와 시스템 설명을 제공하며, 사용자 상태에 따라 적절한 옵션을 표시한다. | |
| 추가 설명 | |
| 시스템의 메인 페이지 역할을 하며, Fraud Detection System에 대한 소개 및 주요 기능에 접근할 수 있는 링크들을 제공한다. | |

# <CSS 파일 분석>

### <style.css>

body {

background: linear-gradient(to right, #0f2027, #203a43, #2c5364);

color: #f1f1f1;

font-family: 'Roboto', sans-serif;

margin: 0;

padding: 0;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

min-height: 100vh;

}

* 설명: 페이지의 기본 배경을 블루톤 그라디언트로 설정하고, 텍스트 색상을 밝은 회색(#f1f1f1)으로 지정하여 명확하게 보이도록 한다. ‘Roboto’ 폰트를 사용하고, 레이아웃을 중앙 정렬하여 사용자 인터페이스가 가운데로 배치되도록 한다.

header {

background-color: rgba(255, 255, 255, 0.05);

width: 100%;

padding: 20px 0;

text-align: center;

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

}

* 설명: 헤더는 투명한 흰색 배경에 그림자를를추가하여 부드러운 효과를 준다. 중앙 정렬로 제목을 표시하고 페이지 상단에 고정한다.

h1 {

font-size: 3rem;

font-weight: 300;

color: #1f8ef1;

text-shadow: 2px 2px 10px rgba(31, 142, 241, 0.4);

}

* 설명: 제목은 큰 글씨체로 표시되며, 밝은 파란색(#1f8ef1)으로 강조되어 시각적 초점이 될 수 있도록 한다. 텍스트 그리마 효과로 부드러운 입체감을 더한다.

nav ul {

list-style: none;

padding: 0;

margin: 0;

display: flex;

justify-content: center;

gap: 20px;

}

nav ul li {

display: inline-block;

}

nav ul li a {

color: #fff;

text-decoration: none;

font-size: 1.2rem;

padding: 10px 20px;

border-radius: 30px;

transition: background-color 0.3s ease, color 0.3s ease;

background: rgba(255, 255, 255, 0.1);

}

nav ul li a:hover {

background-color: #1f8ef1;

color: #fff;

box-shadow: 0 4px 15px rgba(31, 142, 241, 0.6);

}

* 설명: 네비게이션 메뉴는 가운데 정렬되어 있고, 항목 간의 간격이 균등하다. 링크에 마우스를 올렸을 때 색상과 그림자가 변경되면서 시각적 효과를 준다.

form {

background-color: rgba(255, 255, 255, 0.05);

border-radius: 10px;

padding: 30px;

width: 100%;

max-width: 600px;

box-shadow: 0 4px 12px rgba(0, 0, 0, 0.2);

}

form label {

display: block;

margin-bottom: 10px;

color: #f1f1f1;

font-size: 1.2rem;

}

form input[type="file"] {

width: 100%;

padding: 10px;

border-radius: 30px;

border: 2px solid #1f8ef1;

background: rgba(255, 255, 255, 0.1);

color: #fff;

margin-bottom: 20px;

font-size: 1rem;

}

form button {

width: 100%;

padding: 15px;

font-size: 1.2rem;

background-color: #1f8ef1;

border: none;

color: white;

border-radius: 30px;

cursor: pointer;

transition: all 0.3s ease;

}

form button:hover {

background-color: #3f51b5;

box-shadow: 0 8px 25px rgba(63, 81, 181, 0.7);

}

* 설명: 폼은 중앙에 배치되고, 부드러운 그림자 효과와 둥근 모서리로 디자인되어 있다. 버튼은 파란색으로 강조되며, 마우스를 올렸을 때 색상이 더 진해지고 그림자가 추가된다.

table {

width: 100%;

margin-top: 20px;

border-collapse: collapse;

text-align: left;

background-color: rgba(255, 255, 255, 0.05);

border-radius: 10px;

}

th, td {

padding: 15px;

border-bottom: 1px solid rgba(255, 255, 255, 0.2);

color: #f1f1f1;

}

th {

background-color: rgba(31, 142, 241, 0.5);

font-weight: 600;

text-transform: uppercase;

}

tr:hover {

background-color: rgba(255, 255, 255, 0.1);

}

* 설명: 테이블은 데이터를 깔끔하게 정렬하기 위한 요소로, 컬럼 헤더는 파란색으로 강조되고, 테이블 행에 마우스를 올리면 배경색이 부드럽게 변화하여 상호작용을 강화한다.

| 입력 | |
| --- | --- |
| CSS 파일은 프로젝트의 모든 HTML 페이지에 걸쳐 스타일을 적용하며, 사용자가 상호작용하는 폼, 버튼, 테이블 등의 UI 요소에 디자인을 제공한다. | |
| 출력 | |
| 스타일이 적용된 HTML 페이지는 일관된 디자인과 사용자 경험을 제공하며, 시각적 요소들이 정해진 규칙에 따라 렌더링된다. | |
| 매개변수 | |
| 색상 | 다양한 배경 및 텍스트 색상으로 페이지의 시각적 일관성을 유지한다. |
| 폰트 | ‘Roboto’ 폰트를 사용하여 깔끔한 텍스트 표시 |
| 레이아웃 | ‘flexbox’를 이용해 중앙 정렬과 배치 구조를 최적화한다. |
| 예외 처리 | |
| 특별한 예외 처리가 없으며, 모든 스타일이 정상적으로 적용되지 않는 경우 브라우저 기본 스타일이 적용될 수 있다. | |
| 알고리즘 | |
| CSS 스타일은 선언적 언어로 동작하며, 브라우저는 이를 기반으로 페이지 요소를 렌더링한다. 스타일의 순서는 선언된 규칙에 따라 상속되고 적용된다. | |
| 추가 설명 | |
| 이 CSS 파일은 Fraud Detection System의 전체적인 사용자 경험을 고려하여 디자인되었으며, 각 요소가 사용자의 직관적인 인터페이스 상호작용을 돕도록 구성되었다. 각 페이지가 시각적으로 일관성을 유지하면서도 사용자가 쉽게 탐색할 수 있도록 도와준다. | |

# <시각적 요소 (Visual Elements)>

### <와이어프레임 및 시각적 디자인>

* 목적: Fraud Detection System의 주요 페이지를 직관적으로 설계하고, 시각적인 가이드라인을 정의하여 사용자 경험(UX)을 극대화한다. 각 페이지는 사용자가 정보를 쉽게 탐색하고 이해할 수 있도록 구성되며, 시스템의 기능을 명확하게 제공한다.

1. 홈 페이지 (Home Page) 와이어프레임

* 상단 네비게이션:
  + 위치: 페이지 상단에 고정된 네비게이션 바
  + 구성: Home, Upload Data, Dashboard, Login/Logout 링크
* 메인 콘텐츠:
  + 중앙에 시스템 소개 섹션이 있으며, 시스템의 목적과 기능에 대한 설명을 간단히 제공
  + How It Works 섹션: 데이터 업로드부터 이상 거래 탐지 분석까지의 전체 흐름을 시각적으로 설명
* 하단 푸터:
  + 시스템 정보와 저작권을 표시

1. 데이터 업로드 페이지 (Upload Data) 와이어프레임

* 상단 네비게이션:
  + Home, Upload Data, Dashboard, Login/Logout 링크가 포함된 고정 네비게이션
* 업로드 폼:
  + 중앙에 위치한 업로드 폼으로 CSV 파일을 업로드할 수 있는 버튼과 필수 컬럼 안내 메시지가 함께 표시
  + 업로드 버튼 아래에는 에러 메시지가 빨간색으로 표시되어 잘못된 파일 형식이나 오류가 발생할 경우 사용자에게 즉시 피드백을 제공

1. 이상 거래 탐지 대시보드 (Dashboard) 와이어프레임

* 상단 네비게이션:
  + 동일하게 Home, Upload Data, Dashboard, Logout 링크가 포함.
* 중앙 콘텐츠:
  + Kibana 대시보드가 페이지 대부분을 차지하며, 실시간 이상 거래 탐지 결과를 표시하는 차트와 지도가 포함된 iframe이 배치
  + 사용자 데이터를 실시간으로 시각화하여 직관적인 분석을 제공

1. 회원가입 및 로그인 페이지 (Sign Up & Login) 와이어프레임

* 로그인/회원가입 폼:
  + 폼은 페이지 중앙에 배치되며, 사용자명, 비밀번호 입력 필드와 함께 로그인/회원가입 버튼이 포함
  + 오류 메시지는 입력 필드 하단에 배치되어 잘못된 입력에 대한 즉각적인 피드백을 제공한다.
* 회원가입 페이지:
  + 폼에는 추가적으로 이메일 입력 필드가 포함되며, 비밀번호 요구사항이 함께 제공된다.
* 시각적 디자인 가이드라인

1. 컬러 팔레트:

* 배경: 깊은 블루톤 그라디언트 (#0f2027, #203a43, #2c5364)
* 텍스트: 밝은 회색 (#f1f1f1) 및 강조 색상으로 파란색 (#1f8ef1)
* 버튼: 기본 파란색 (#1f8ef1) 및 마우스 오버 시 진한 파란색 (#3f51b5)

1. 폰트 스타일:

* 기본 폰트: ‘Roboto’, sans-serif
* 제목: 대형 글씨체(3rem), 얇은 두께(300)로 주요 섹션 제목을 시각적으로 강조

1. 버튼 및 상호작용 요소

* 버튼과 링크는 직관적으로 설계되어 있으며, 마우스를 올릴 때 배경색이 변화하고 그림자가 추가되어 상호작용의 피드백을 제공한다.

1. 레이아웃:

* 모든 페이지는 중앙 정렬되어 있고, flexbox 레이아웃을 사용하여 화면 크기에 맞춰 자동 조정된다.

1. 이미지 및 테이블 시각화:

* 이미지 시각화는 둥근 모서리와 그림자 효과가 적용되어, 사용자에게 친숙하고 시각적으로 편안한 환경을 제공한다.
* 테이블은 명확한 구분선과 함께 시각적으로 구분하기 쉽도록 디자인되었다.

<색상 팔레트 (Color Palette)>

* 설명: 프로젝트의 색상 팔레트는 사용자 인터페이스의 일관성과 가독성을 높이고, 시각적 요소들이 잘 조화를 이루도록 구성되었다.

1. 기본 색상 (Primary Colors):

* #0f2027 (Deep Blue Black)
* #203a43 (Dark Teal)
* #2c5364 (Blue Gray)
* 설명: 배경에는 깊은 블루톤 그라디언트가 사용되어 안정적이고 전문적인 느낌을 준다. 이 색상 조합은 시스템의 신뢰성을 강조한다.

1. 텍스트 색상 (Text Colors):

* 기본 텍스트: #f1f1f1 (Light Gray)
  + 설명: 흰색에 가까운 회색으로, 어두운 배경과 대비되어 높은 가독성을 제공한다.
* 제목 강조색: #1f8ef1 (Light Blue)
  + 설명: 주요 제목과 링크는 밝은 파란색으로 강조되어 중요한 정보가 눈에 잘 띄도록 한다.

1. 버튼 및 링크 색상 (Buttons & Links):

* 기본 버튼 색상: #f8ef1 (Light Blue)
  + 설명: 주요 버튼 및 상호작용 요소에 사용되어 시각적으로 눈에 잘 띄며, 파란색은 신뢰와 안전을 상징한다.
* 호버 상태 색상: #3f51b5 (Dark Blue)
  + 설명: 사용자가 버튼 위에 마우스를 올렸을 때 색상이 더 진한 파란색으로 변화하여 상호작용을 시각적으로 강화한다.

1. 보조 색상 (Secondary Colors):

* 에러 메시지: #ff0000 (Red)
  + 설명: 에러 메시지는 강한 빨간색을 사용하여 사용자에게 명확한 경고 신호를 제공한다.
* 그림자 효과: rgba(0, 0, 0, 0.4) (Black with Transparency)
  + 설명: 그림자 효과는 시각적으로 요소들이 떠 있는 느낌을 주어 페이지의 깊이를 더한다.

<폰트 (Typography)>

* 설명: 폰트 선택은 가독성과 디자인의 일관성을 고려하여 이루어졌다. Fraud Detection System의 디자인에서 사용된 주요 폰트는 다음과 같다.

1. 기본 폰트:

* Roboto (Sans-Serif)
  + 설명: ‘Roboto’ 폰트는 깔끔하고 모던한 느낌을 주며, 가독성이 높아 다양한 화면 크기에서 잘 어울린다. 특히, 전문적인 분위기를 유지하면서도 현대적인 느낌을 준다.

1. 폰트 크기 (Font Sizes):

* 제목 (h1): 3rem
  + 설명: 주요 제목은 큰 폰트 크기를 사용하여 페이지의 핵심 정보를 강조한다.
* 기본 텍스트: 1rem
  + 설명: 일반 텍스트는 1rem으로 설정되어, 가독성을 유지하면서 페이지의 다른 시각적 요소와 잘 어울린다.

1. 폰트 두께 (Font Weights):

* 제목: 300 (Light)
  + 설명: 제목은 얇은 두께를 사용하여 시각적으로 가벼운 느낌을 주며, 전체적인 페이지의 디자인과 조화를 이룬다.
* 본문 텍스트: 400 (Regular)
  + 설명: 기본 텍스트는 적당한 두께로 설정되어 가독성을 유지한다.

# <사용자 경험 흐름 (UX Flow)>

### <로그인 (Login Flow)>

1. 로그인 페이지 접근:

* 사용자는 홈 페이지의 로그인 버튼을 클릭하여 로그인 페이지로 이동한다.

1. 사용자 인증:

* 로그인 폼에서 사용자명과 비밀번호를 입력한다. 입력된 정보가 Django Authentication 시스템을 통해 인증된다.
* 인증 성공 시, 사용자에게 환영 메시지가 표시되고, 대시보드 또는 홈 페이지로 자동 리다이렉트된다.
* 인증 실패 시, “Invalid username or password”와 같은 에러 메시지가 빨간색으로 표시되어 즉각적인 피드백을 제공한다.

1. 추가 정보:

* 로그인 페이지 하단에는 회원가입 링크가 있어, 계정이 없는 사용자는 바로 회원가입을 진행할 수 있다.

### <데이터 업로드 (Data Upload Flow)>

1. 업로드 페이지 접근:

* 사용자는 네비게이션 바에서 Upload Data 버튼을 클릭하여 데이터 업로드 페이지로 이동한다.

1. CSV 파일 선택 및 업로드:

* 페이지 중앙에 위치한 업로드 폼에서 CSV 파일을 선택한다.
* CSV 파일에는 필수 컬럼인 ‘Amount’, ‘Customer ID’, ‘Merchant ID’, ‘Lat’, ‘Long’가 포함되어 있어야 한다.
* 파일이 선택되면 업로드 버튼을 클릭하여 서버로 전송된다.

1. 파일 처리 및 업로드 성공:

* 업로드된 파일은 서버에서 처리되어 세션에 저장된다.
* 사용자가 업로드한 데이터는 MySQL 데이터베이스에 저장되고, 시스템에서 처리될 준비가 된다.
* 업로드 후 사용자는 대시보드 시각화 페이지로 리다이렉트된다.

1. 오류 처리:

* 만약 CSV 파일 형식이 잘못되었거나 필수 컬럼이 누락된 경우, 에러 메시지가 빨간색으로 표시되어 사용자에게 오류를 알린다.

### <대시보드 시각화 과정 (Dashboard Visualization Flow)>

1. 대시보드 접근:

* 사용자는 업로드한 데이터를 시각화하기 위해 대시보드 페이지로 리다이렉트된다.
* 대시보드에서는 Kibana 대시보드가 iframe을 통해 임베드되어 실시간 데이터 시각화를 제공한다.

1. 실시간 이상 거래 탐지 데이터 시각화

* 사용자는 업로드한 거래 데이터를 기반으로 이상 거래 탐지 결과를 지도와 차트로 확인할 수 있다.
* Kibana 대시보드는 Elasticsearch에서 수집된 데이터를 실시간으로 시각화하며, 사용자 ID 필터를 적용하여 사용자 개인의 데이터만 확인할 수 있다.

1. 사용자 인터랙션:

* 사용자는 대시보드에서 필터링 옵션이나 시간 범위 설정을 조정하여, 원하는 데이터 범위를 분석할 수 있다.
* Kibana를 통해 실시간으로 업데이트되는 차트와 지표를 통해 데이터 흐름을 명확하게 시각적으로 파악할 수 있다.

1. 이상 거래 탐지 및 알림:

* 대시보드에서 이상 거래가 감지되면, 시스템이 자동으로 사용자의 이메일로 이상 거래 발생 알림을 발송한다.
* 사용자에게는 사기 거래와 관련된 거래 ID, 금액, 고객 및 상점 정보가 포함된 상세 내용을 확인할 수 있는 이메일이 전송된다.

1. 추가 시각화:

* 만약 사용자가 대시보드 내에서 시각화된 데이터를 다운로드하고 싶다면, 제공된 버튼을 통해 PNG 이미지나 CSV 파일 형식으로 다운로드할 수 있다.

### <이메일 알림 기능>

1. 이상 거래 탐지:

* 사용자가 업로드한 데이터를 기반으로 이상 거래 탐지 모델(Isolation Forest)이 데이터를 분석하고, 이상 거래를 감지한다.
* 대시보드에 시각화된 데이터를 통해 사용자는 실시간으로 이상 거래를 확인할 수 있다.

1. 이메일 알림 조건:

* 이상거래 탐지 시스템은 이상 거래가 1건 이상 감지될 경우, 자동으로 이메일 알림을 전송한다.
* 이상 거래가 감지되지 않을 경우에는 이메일 알림이 전송되지 않으며, 정상 거래로 표시된다.

1. 이메일 알림 발송 과정

* 이상 거래가 발생하면 시스템이 이메일 알림을 자동 생성한다.
* 이메일 내용은 시스템 내 템플릿(fraud/email\_template.html)을 사용하여 생서오디며, 이상 거래와 관련된 상세 정보를 포함한다.
* 이메일 템플릿에는 이상 거래 정보(거래 금액, 고객 ID, 상점 ID, 위도/경도 정보 등)가 담긴다.

1. 이메일 수신자:

* 이메일은 해당 사용자의 계쩡에 등록된 이메일 주소로 발송된다.
* Django의 이메일 백엔드 설정을 사용하여 SMTP 서버를 통해 발송된다. (SMTP 서버 설정은 settings.py에 미리 구성되어 있음)

1. 이메일 발송 성공 여부:

* 시스템은 이메일 발송이 성공적으로 완료되었는지 여부를 확인한다. 발송 실패 시, 서버 로그에 실패 메시지가 기록되며, 실패 원인이 표시된다.
* 이메일이 정상적으로 발송되었을 경우, 사용자에게 전송된 이메일 내용에는 이상 거래 상세 내역이 포함되어 있다.

1. 이메일 내용:

* 이메일 제목: [Sniffers] Fraudulent Transaction Detected!
* 이메일 본문에는 다음과 같은 내용이 포함된다.
  + 거래 금액 (Amount)
  + 고객 ID (Customer ID)
  + 상점 ID (Merchant ID)
  + 위도 및 경도 (Latitude & Longitude)
  + 거래가 사기 거래로 감지되었다는 알림 메시지
* HTML 포맷으로 발송되며, 이메일 템플릿이 사용된다. 만약 HTML 지원이 되지 않는 경우 텍스트 포맷으로도 내용을 전송한다.

1. 사용자 피드백:

* 사용자는 이메일을 통해 즉각적으로 이상 거래 발생 사실을 인지할 수 있으며, 이메일 내의 정보를 바탕으로 필요한 조치를 취할 수 있다.
* 이메일에 명확한 거래 정보와 이상 거래의 증거가 포함되어 있어, 사용자는 추가 조사를 하거나 시스템에서 데이터를 검토할 수 있다.

# <접근성 고려사항>

### <색상 대비 (Color Contrast)>

* 설명: Fraud Detection System에서는 모든 사용자가 시스템을 원활하게 이용할 수 있도록 웹 접근성 표준을 준수하여 색상 대비를 고려했다. 색상 대비는 시각적으로 인식하기 어려운 사용자, 특히 색약 또는 시력이 저하된 사용자를 위해 중요한 요소이다.
* 색상 대비 기준

1. WCAG (Web Content Accessibility Guidelines) 2.1에 따르면, 텍스트와 배경간의 명암비(Contrast Ratio)는 다음을 충족해야 한다.

* 최소 명암비: 일반 텍스트는 4.5:1, 굵은 텍스트는 3:1 이상의 명암비를 유지해야 한다.
* 큰 텍스트(18pt 이상, 굵은 텍스트는 14pt 이상)는 최소 3:1의 명암비를 가져야 한다.
* Fraud Detection System의 색상 대비 적용

1. 텍스트 대비:

* 기본 텍스트 색상은 #f1f1f1 (밝은 회색), 배경은 #0f2027 (어두운 블루톤)으로 구성되어, 명암비가 15.1:1로 WCAG 기준을 크게 초과한다. 이 대비는 시각적 장애가 있는 사용자도 명확하게 텍스트를 읽을 수 있게 도와준다.
* 제목 및 강조된 텍스트는 #1f8ef1 (밝은 파랑)으로 설정되며, 배경과의 명암비가 8.59:1로 높은 가독성을 제공한다.

1. 버튼 및 링크:

* 버튼과 링크는 기본적으로 #1f8ef1 (밝은 파랑)이며, 배경과의 대비는 8.59:1로 높은 가독성을 제공한다.
* 버튼이나 링크에 마우스를 올렸을 때 호버 색상은 #3f51b5 (어두운 파랑)으로 변경되어, 배경과의 명암비가 4.53:1로 접근성을 유지하면서 상호작용을 시각적으로 강조한다.

1. 에러 메시지:

* 에러 메시지는 #ff0000 (빨강)으로 표시되며, 배경과의 명암비가 5.25:1로 시각적으로 명확하게 인지할 수 있다.
* 에러 메시지가 사용자에게 즉각적으로 주의를 줄 수 있도록 대비가 높게 설정되어 있다.

1. 배경과의 대비:

* 배경은 깊은 블루톤 그라디언트로 설정되어 있으며, 모든 텍스트와 버튼은 배경과 높은 대비를 유지하여 가독성을 높였다.
* 특히, 주요 정보가 제공되는 본문 영역에서는 배경과 텍스트 사이의 대비를 극대화하여 시각적 장애가 있는 사용자가 정보를 쉽게 인식할 수 있도록 했다.
* 고려사항 및 테스트 결과
  + 명암비 테스트 도구를 사용하여 페이지 전반의 텍스트와 배경 간의 명암비를 확인했으며, WCAG 2.1 기준의 AA 및 AAA 레벨을 만족하는 명암비를 유지하고 있다.
  + 페이지 전반에 걸친 색상 선택은 시각 장애나 색약을 가진 사용자가 텍스트와 상호작용 요소를 명확하게 인지할 수 있도록 디자인되었다.
  + 버튼 및 링크는 색상 변경뿐만 아니라, 그림자 효과를 추가하여 사용자가 마우스를 올렸을 때 상호작용이 일어나고 있음을 명확하게 전달한다.

### <텍스트 크기 조정 가능 여부>

* 설명: Fraud Detection System은 모든 사용자가 화면의 텍스트 크기를 조정하여 쉽게 읽을 수 있도록 텍스트 크기 조정 기능을 고려하여 설계되었다. 이 기능은 특히 시각적으로 불편함을 느끼는 사용자에게 중요하며, 웹 접근성 기준을 충족하는 필수 요소 중 하나이다.
* WCAG 기준
  + WCAG 2.1 가이드라인에 따르면, 사용자는 브라우저의 기본 텍스트 크기 설정을 변경함으로써 페이지 내 텍스트를 최대 200%까지 확대할 수 있어야 하며, 이 때도 페이지 콘텐츠가 깨지거나 가독성이 저하되지 않아야 한다.
  + 또한, 텍스트 크기를 조정해도 콘텐츠가 잘리거나 레이아웃이 깨지지 않아야 한다.
* Fraud Detection System의 텍스트 크기 조정 고려사항

1. 상대적인 단위 사용:

* 모든 텍스트 크기는 상대적 단위인 ‘rem’ 또는 ‘em’을 사용하여 설정되었다. 이를 통해 사용자가 브라우저 설정에서 텍스트 크기를 조정하면, 시스템에서 제공되는 모든 텍스트가 동일한 비율로 확대 또는 축소된다.
* 예를 들어, 제목(h1)은 3rem, 기본 텍스트는 1rem으로 설정되어 있어, 사용자가 텍스트 크기를 조정할 때 전체 콘텐츠가 비율에 맞게 확대되거나 축소된다.

1. 페이지 레이아웃 유연성:

* 시스템의 전체 레이아웃은 유연한 레이아웃을 적용하여, 텍스트 크기를 확대해도 레이아웃이 깨지지 않도록 설계되었다.
* 특히, form 요소, 버튼, 네비게이션 메뉴 등 주요 상호 작용 요소들이 텍스트 크기에 맞춰 자동으로 크기가 조정되어, 사용자 경험을 저해하지 않고 일관된 UI를 제공한다.

1. 브라우저 텍스트 확대 기능 지원:

* 대부분의 브라우저는 기본적으로 텍스트 크기를 조정할 수 있는 기능을 제공한다. Fraud Detection System은 이러한 브라우저의 확대/축소 기능을 완전히 지원한다.
* 사용자가 브라우저에서 텍스트 크기를 200%까지 확대해도, 페이지의 내용이 손상되지 않으며, 모든 텍스트는 명확하고 읽기 쉽게 유지된다.

1. 반응형 디자인 적용:

* 시스템은 반응형 디자인을 적용하여, 다양한 화면 크기와 해상도에서도 텍스트 크기가 자동으로 조정된다.
* 모바일이나 태블릿에서 텍스트 크기를 크게 설정한 경우에도 페이지의 콘텐츠가 스크롤 없이 자연스럽게 배치되도록 설계되었다.
* 구체적인 적용 사례
  + 헤더 및 본문 텍스트:
    - 헤더(h1, h2, h3)와 본문 텍스트(p)는 각각 3rem, 2rem, 1rem 등으로 상대적 크기를 사용하여 설정되었다. 사용자가 텍스트 크기를 조정하면 헤더와 본문 텍스트 크기도 비례적으로 조정된다.
* 폼 필드 및 버튼:
  + 폼 필드와 버튼의 텍스트 크기는 1rem으로 설정되어, 텍스트 크기 조정 시 버튼의 크기와 폼 필드의 높이도 자동으로 조정된다. 이렇게 하면 텍스트가 잘리거나 버튼의 내용이 보이지 않는 문제가 발생하지 않는다.
* 테스트 결과
  + 텍스트 크기 200% 확대: 브라우저에서 텍스트 크기를 200%로 확대해도 모든 텍스트는 여전히 명확하게 표시되며, 페이지의 다른 시각적 요소들과 충돌하지 않는다.
  + 크기 조정 중 레이아웃 깨짐 없음: 텍스트 크기를 조정하더라도 페이지 레이아웃이 깨지거나 요소들이 겹치는 현상은 발생하지 않는다. 모든 콘텐츠가 깔끔하게 정렬된다.

### < 스크린 리더 호환성 (Screen Reader Compatibility)>

* 설명: Fraud Detection System은 시각 장애인 또는 저시력 사용자가 웹 콘텐츠를 효과적으로 탐색하고 이해할 수 있도록 스크린 리더와의 호환성을 고려하여 설계되었다. 스크린 리더는 웹 페이지의 텍스트와 인터페이스 요소를 음성으로 변환하여 사용자가 웹 사이트의 정보를 청취할 수 있게 해주는 중요한 접근성 도구이다.
* WCAG 기준
  + WCAG 2.1의 접근성 가이드라인에 따르면, 스크린 리더가 웹 페이지 내 모든 중요한 정보와 상호작용 가능한 요소를 정확히 인식하고 전달할 수 있어야 한다.
  + 웹 페이지의 모든 인터랙티브 요소는 스크린 리더에 의해 인식 가능해야 하며, 적절한 대체 텍스트와 ARIA 레이블을 제공해야 한다.
* Fraud Detection System의 스크린 리더 호환성 적용사항

1. HTML 구조 및 시멘틱 요소 사용:

* 웹 페이지는 시멘틱 HTML을 사용하여 구조화되어 있다. 시멘틱 태그(예: <header>, <nav>, <main>, <footer> 등)를 사용하여 페이지 내의 각 섹션을 명확하게 구분하고, 스크린 리더가 이 구조를 올바르게 전달할 수 있도록 했다.
* <h1> 태그는 페이지의 주요 제목으로 사용되며, 하위 제목은 <h2>, <h3> 등의 태그를 사용하여 시각적으로나 기능적으로 스크린 리더가 페이지 내 계층 구조를 이해하도록 돕는다.

1. ARIA 레이블 사용:

* ARIA 속성(aria-label, aria-labelledby, role 등)을 사용하여, 스크린 리더가 제대로 인식할 수 없는 요소에 대해 명확한 설명을 제공하고 있다.
* 예를 들어, 네비게이션 메뉴는 ‘role = “navigation”’ 속성을 사용하여 스크린 리더가 이것이 페이지의 네비게이션 영역임을 알 수 있도록 한다.
* 버튼과 링크에는 명확한 레이블을 제공하여, 스크린 리더가 버튼이나 링크의 기능을 정확히 설명할 수 있게 했다. 예를 들어, “Upload Data” 버튼은 ‘aria-label = “Upload Transcation Data”’와 같은 속성을 사용하여 그 목적을 명확히 전달한다.

1. 대체 텍스트 (Alt Text) 제공:

* 페이지 내 모든 이미지에는 적절한 대체 텍스트(alt text)가 제공되어, 스크린 리더가 이미지를 설명할 수 있다.
* 대체 텍스트는 이미지의 내용이나 기능을 간단하고 명확하게 설명하여, 이미지가 제공하는 정보를 놓치지 않도록 한다. 예를 들어, 거래 지도 이미지는 ‘alt = “Fraud Prediction Map”’으로 설정되어, 시각 장애 사용자도 해당 이미지의 의미를 파악할 수 있게 돕는다.

1. 폼 요소와 레이블 연결:

* 모든 폼 요소(input, button, select 등)는 label 태그와 올바르게 연결되어, 스크린 리더가 해당 필드의 목적을 설명할 수 있다.
* 예를 들어, 파일 업로드 폼에서 CSV 파일 선택 필드는 ‘label for = “file”’과 연결되어, 스크린 리더가 “Choose a CSV file to upload”라는 메시지를 사용자에게 읽어준다.

1. 키보드 내비게이션 지원:

* 스크린 리더 사용자는 일반적으로 키보드 내비게이션을 통해 웹 페이지를 탐색한다. Fraud Detection System은 모든 상호작용 가능한 요소들이 키보드로 접근 가능하도록 설계되었다.
* 링크, 버튼, 폼 필드 등은 Tab 키로 쉽게 이동할 수 있으며, Enter 키로 상호작용할 수 있다.

1. 동적 콘텐츠의 스크린 리더 대응:

* 시스템에서 발생하는 동적 콘텐츠(예: 거래 업로드 후 알림, 이상 거래 감지 후 이메일 발송)는 스크린 리더에 의해 인식되도록 설계되었다.
* 예를 들어, CSV 파일 업로드 오류가 발생하면, 스크린 리더는 “The CSV file format is incorrect. Please include ‘Amount’, ‘Customer ID’, ‘Merchant ID’, ‘Lat’, and ‘Long’ columns.”와 같은 오류 메시지를 즉시 읽어준다.
* 구체적인 적용 사례
  + 네비게이션 메뉴:
    - 네비게이션 메뉴의 각 링크에 aria-label을 추가하여, 스크린 리더가 “Go to Home”, “Upload Data”, “View Dashboard in Kibana”와 같은 메시지를 전달할 수 있게 했다.
* 거래 업로드 및 대시보드:
  + 사용자가 거래 데이터를 업로드할 때, 업로드 과정에서 발생하는 모든 피드백(성공, 오류, 필수 컬럼 누락 등)은 스크린 리더를 통해 사용자에게 음성으로 전달된다.
  + 대시보드에서 데이터가 시각화되는 과정은 스크린 리더에 의해 적절히 설명될 수 있도록 각 요소에 역할 및 레이블이 제공된다.
* 테스트 결과
  + 스크린 리더 테스트:
    - NVDA (Windows) 및 VoiceOver (Mac)와 같은 스크린 리더 소프트웨어를 사용한 테스트 결과, 모든 텍스트와 상호작용 가능한 요소가 제대로 읽히고 설명되었다.
    - 스크린 리더는 모든 레이블, 폼 필드, 버튼에 대한 명확한 설명을 제공하며, 오류 메시지 및 성공 메시지도 올바르게 전달된다.