# Asthma Tracking and Management System — Project Documentation

“Track. Predict. Breathe Better.”

## 1. Introduction ✨

### 1.1 Purpose

This document defines the requirements for the Asthma Tracking and Management System, a web-based platform that helps users track inhaler usage, monitor environmental factors, and analyze asthma risk patterns. The system connects to a smart inhaler device via Bluetooth and visualizes the data in an interactive dashboard. It also provides alerts when inhaler use exceeds safe limits and offers preventive recommendations.

### 1.2 Product Scope

The system will:  
- Connect securely with the user’s inhaler device using Bluetooth.  
- Record inhaler usage, time, temperature, and location.  
- Display usage patterns and environmental data through easy-to-read charts.  
- Use predictive analysis (via AI models) to suggest preventive measures.  
- Send real-time alerts to users when daily or weekly limits are exceeded.  
- Provide a simple dashboard for patients and doctors to monitor asthma activity.  
  
The system does not control or deliver medication, replace a doctor’s advice, or act as emergency medical support.

## 2. Overall Description 🗺️

### 2.1 Product Perspective

This is a full-stack web system built using modern technologies. It integrates:  
- Frontend (React + HTML + CSS + JavaScript): For user interaction and real-time data visualization.  
- Backend (Flask + Express.js): For data processing, API communication, and AI predictions.  
- Database (MySQL): For storing inhaler usage and environment data.  
- External APIs: For weather, temperature, and air quality information.

### 2.2 User Types and Characteristics

|  |  |  |
| --- | --- | --- |
| User Type | Description | Key Need |
| Asthma Patient (User) | Regularly uses inhaler and monitors health | Simple interface, timely alerts, visual data |
| Doctor/Clinician | Reviews patient data and patterns | Downloadable reports, trend insights |
| System Admin | Manages users, backend services | System uptime and secure data management |

## 3. Functional Requirements ⚙️

### 3.1 Data Collection and Synchronization

FR-01: Record every inhaler use with date and time.  
FR-02: Capture GPS location during inhaler usage.  
FR-03: Get air quality and temperature data from external APIs.  
FR-04: Track remaining inhaler doses (SABA medication).

### 3.2 AI and Prediction Features

FR-05: Predict asthma risk based on inhaler usage and weather patterns.  
FR-06: Alert users to avoid outdoor activities if air quality is poor.  
FR-07: Remind users to take preventive medication when risk increases.

### 3.3 Reporting and Alerts

FR-08: Show live color-coded risk status (Green/Yellow/Red).  
FR-09: Create monthly PDF reports for doctors and patients.  
FR-10: Send alert messages when inhaler usage exceeds safe limits.

## 4. System Architecture 🏗️

### 4.1 Data Flow

Smart Inhaler → Frontend (React App) → Backend (Flask + Express APIs) → MySQL Database → AI Engine (Python) → Dashboard + Alerts

### 4.2 Components

- Frontend: React-based responsive interface.  
- Backend: Flask (AI logic) + Express (API handling).  
- Database: MySQL for storing usage and environment data.  
- External APIs: Weather and Air Quality for contextual insights.

## 5. Non-Functional Requirements 🛡️

### 5.1 Security and Privacy

- All user data must be encrypted before storage.  
- Authentication must be required for every login session.  
- Only authorized users can access or edit medical records.

### 5.2 Performance and Reliability

- The system should respond to any request within 1 second.  
- Uptime should be at least 99%.  
- Should support multiple users without slowing down.

### 5.3 Usability

- The dashboard must be simple and accessible to all age groups.  
- Risk level should be clearly indicated using color-coded visuals.  
- Users should understand their status within 30 seconds.

## 6. Technologies Used 💻

Frontend: HTML, CSS, JavaScript, React  
Backend: Flask, Express.js  
Database: MySQL  
AI Logic: Python (for risk prediction and analytics)  
Version Control: Git + GitHub