**Enhanced Prescription Analyzer - Project Documentation**

**Project Overview**

The Enhanced Prescription Analyzer is a web-based application that analyzes medical prescriptions to provide dosage recommendations, alternative medications, and clinical insights based on patient demographics and medical conditions. The system uses fuzzy string matching and clinical decision-making algorithms to assist healthcare professionals and patients in understanding medication appropriateness.

**Key Features**

* **Drug Entity Extraction**: Automatically identifies medications from prescription text using fuzzy matching algorithms
* **Age-Appropriate Dosing**: Calculates pediatric and adult dosages based on patient age and weight
* **Disease-Drug Correlation**: Analyzes medication relevance for specific medical conditions
* **Alternative Medication Suggestions**: Provides therapeutic alternatives for identified drugs
* **Interactive Web Interface**: User-friendly form-based interface for prescription analysis

**Technical Architecture**

**Technology Stack**

* **Backend Framework**: FastAPI
* **String Matching**: RapidFuzz library for fuzzy text matching
* **Web Server**: Uvicorn with asyncio support
* **Frontend**: HTML/CSS with responsive design
* **Deployment**: Ngrok tunneling for public access
* **Environment**: Python with nest\_asyncio for Jupyter/Colab compatibility

**System Components**

**1. Drug Database (DRUG\_DB)**

A comprehensive dictionary containing:

* Drug names and aliases
* Adult and pediatric dosing formulations
* Clinical indications
* Safety notes and contraindications
* Alternative medication suggestions

**2. Core Algorithms**

**Fuzzy Drug Matching**

python

def fuzzy\_match\_drug(name: str, limit=1):

*# Uses RapidFuzz WRatio scorer with 70% confidence threshold*

*# Maps common names and aliases to standardized drug entries*

**Drug Entity Extraction**

python

def extract\_drug\_entities(text: str):

*# Parses prescription text and identifies unique medications*

*# Handles punctuation and formatting variations*

**Clinical Analysis Engine**

python

def analyze\_drugs(drug\_entities, age\_years, weight\_kg, disease):

*# Calculates age and weight-appropriate dosing*

*# Evaluates drug-disease correlation*

*# Generates clinical recommendations*

**API Endpoints**

**GET /**

* **Purpose**: Serves the main prescription input form
* **Response**: HTML form with patient demographics and prescription text fields
* **Features**: Responsive design with input validation

**POST /analyze**

* **Purpose**: Processes prescription analysis requests
* **Parameters**:
  + prescription\_text: Free-text prescription input
  + age: Patient age in years
  + weight: Patient weight in kg (optional)
  + disease: Primary medical condition (optional)
* **Response**: Formatted HTML results page with analysis table

**Drug Database Schema**

Each drug entry contains:

python

{

"drug\_name": {

"aliases": ["list", "of", "alternative", "names"],

"adult\_dose": "Adult dosing instructions",

"child\_dose\_text": "Pediatric dosing description",

"child\_dose\_kg\_mg": (min\_mg\_per\_kg, max\_mg\_per\_kg),

"indications": ["list", "of", "medical", "conditions"],

"notes": "Clinical notes and warnings",

"alternatives": ["alternative", "medications"]

}

}

**Currently Supported Medications**

1. **Paracetamol/Acetaminophen**
   * Aliases: Calpol, Tylenol
   * Indications: Fever, pain, headache, migraine
   * Pediatric: 10-15 mg/kg per dose
2. **Ibuprofen**
   * Aliases: Motrin, Advil, Nurofen
   * Indications: Fever, pain, inflammation, arthritis
   * Pediatric: 5-10 mg/kg every 6-8 hours
3. **Amoxicillin**
   * Aliases: Amoxil
   * Indications: Bacterial infections, pneumonia
   * Pediatric: 40-90 mg/kg/day divided doses
4. **Azithromycin**
   * Aliases: Zithromax, Z-Pak
   * Indications: Bacterial infections, pneumonia
   * Special dosing: Loading dose followed by maintenance
5. **Loratadine**
   * Aliases: Claritin
   * Indications: Allergies, hay fever, hives
   * Age-based dosing (not weight-based)
6. **Metoprolol**
   * Aliases: Betaloc, Lopressor
   * Indications: Hypertension, angina, heart failure
   * Adult-only medication

**Clinical Decision Logic**

**Dosage Calculations**

**Pediatric Patients (Age < 18)**

* Weight-based calculations using mg/kg formulas
* Age-appropriate formulations when weight unavailable
* Safety considerations for pediatric populations

**Adult Patients (Age ≥ 18)**

* Standard adult dosing regimens
* Maximum daily dose limits
* Frequency and timing recommendations

**Drug-Disease Correlation**

The system evaluates medication appropriateness by:

* Matching drug indications with patient conditions
* Providing relevance indicators (✅ Relevant, ⚠ May not be primary treatment)
* Including clinical notes and contraindications

**Alternative Medication Logic**

* Suggests therapeutically equivalent alternatives
* Considers different drug classes for similar indications
* Provides options for patients with contraindications

**User Interface Design**

**Main Form Features**

* **Prescription Text Area**: Large text input for free-form prescription entry
* **Patient Demographics**: Age and weight input fields
* **Medical Condition**: Optional disease/condition specification
* **Responsive Grid Layout**: Optimized for various screen sizes
* **Professional Styling**: Healthcare-appropriate color scheme and typography

**Results Display**

* **Patient Summary**: Key demographics and condition
* **Analysis Table**: Drug-by-drug breakdown with dosing recommendations
* **Clinical Notes**: Relevant warnings and considerations
* **Alternative Medications**: Therapeutic substitution options
* **Medical Disclaimer**: Clear indication that this is educational only

**Installation and Deployment**

**Dependencies**

bash

pip install fastapi uvicorn pyngrok requests rapidfuzz nest\_asyncio

**Configuration**

1. **Ngrok Setup**: Requires authentication token for public URL generation
2. **Environment**: Compatible with Jupyter/Colab environments
3. **Port Configuration**: Default port 8000 with HTTP tunnel

**Deployment Process**

python

*# 1. Install dependencies automatically*

*# 2. Configure ngrok authentication*

*# 3. Create public tunnel*

*# 4. Start FastAPI server in background thread*

*# 5. Provide public URL for access*

**Safety and Limitations**

**Built-in Safety Features**

* **Medical Disclaimers**: Prominent warnings that this is educational only
* **Professional Consultation**: Clear recommendations to consult healthcare providers
* **Limited Scope**: Focuses on common medications with well-established dosing

**System Limitations**

* **Database Scope**: Limited to manually curated drug entries
* **Clinical Complexity**: Cannot account for all patient-specific factors
* **Drug Interactions**: Does not check for drug-drug interactions
* **Real-time Updates**: Database requires manual updates for new medications

**Recommended Use Cases**

* **Educational Tool**: For learning about medication dosing principles
* **Clinical Reference**: Quick lookup for common medications
* **Decision Support**: Preliminary analysis requiring professional validation

**Future Enhancement Opportunities**

**Database Expansion**

* Integration with pharmaceutical APIs (RxNorm, FDA datasets)
* Real-time drug information updates
* Expanded medication coverage

**Clinical Features**

* Drug-drug interaction checking
* Allergy and contraindication screening
* Laboratory value considerations (renal function, liver function)

**Advanced Analytics**

* Machine learning for prescription pattern analysis
* Natural language processing for complex prescription parsing
* Integration with electronic health records

**User Experience**

* Mobile application development
* Voice input capabilities
* Multi-language support
* Healthcare provider dashboard

**Technical Implementation Notes**

**Asyncio Compatibility**

The application uses nest\_asyncio.apply() to ensure compatibility with Jupyter/Colab environments where asyncio loops may already be running.

**String Matching Algorithm**

RapidFuzz WRatio scoring provides robust matching against:

* Spelling variations and typos
* Brand name vs. generic name differences
* Common abbreviations and shorthand

**Error Handling**

* Graceful degradation for unrecognized medications
* Input validation for numeric fields
* Clear error messages for malformed requests

**Performance Considerations**

* In-memory drug database for fast lookup
* Efficient fuzzy matching with configurable thresholds
* Minimal external API dependencies

**Conclusion**

The Enhanced Prescription Analyzer demonstrates a practical application of clinical informatics principles, combining drug knowledge bases with algorithmic decision support. While designed as an educational and reference tool, it showcases the potential for technology to assist in medication management and clinical decision-making.

The modular architecture allows for straightforward expansion and integration with more comprehensive healthcare systems, making it a valuable foundation for future development in digital health applications.