

System Architecture Documentation

1. System Architecture

1.1 Framework and Components

- Built using Flask, a lightweight Python web framework
- Three main components:
 - Frontend: HTML templates (index.html, details.html, result.html)
 - Backend: Flask application (app.py)
 - Knowledge Base: JSON file (knowledge_base.json)

1.2 Application Flow

1. User selects symptoms on the index page
2. User provides symptom details (duration, severity) on details page
3. System evaluates symptoms using forward chaining inference
4. Results displayed on result page

1.3 Data Flow

- User input → Session storage → Rule evaluation → Result generation

2. Key Design Decisions

2.1 Session Management

- Uses Flask's session management to store:
 - Selected symptoms
 - User-provided symptom details
- Enables multi-step interaction without database

2.2 Rule Evaluation System

- Forward chaining inference engine
- Supports complex conditions:
 - Symptom presence
 - Minimum duration
 - Required severity
 - Nested OR logic

2.3 Knowledge Base Structure

- JSON format for easy maintenance
- Contains:
 - List of symptoms
 - Rules with conditions and recommendations

2.4 Error Handling

- Basic validation:
 - At least one symptom selected
 - All fields completed in details form

3. Known Limitations

3.1 Scalability

- Current implementation uses in-memory session storage
- Not suitable for high traffic or distributed deployment

3.2 Knowledge Base

- Static JSON file requires manual updates
- Limited to predefined rules and symptoms

3.3 Security

- Basic session management
- No user authentication
- No data encryption

3.4 Error Handling

- Limited validation
- No logging or monitoring
- Basic error messages

3.5 Performance

- Rule evaluation is linear $O(n)$
- No caching of results
- No optimization for large knowledge bases