Software Requirements Documentation (SRD)

Project Title: TriVita Advance Medical Event Prediction System

Version: 1.0

Prepared By: Astika Sinha, Dhruv Sharma, Ayushmaan Tyagi

# 1. Objective

To build a machine learning-based system that classifies medical symptom descriptions into categories and generates a detailed PDF report highlighting the prediction and key contributing keywords.

# 2. Functional Requirements

|  |  |
| --- | --- |
| ID | Requirement Description |
| FR1 | The system shall accept free-text symptom descriptions from the user. |
| FR2 | The system shall preprocess the text using TF-IDF vectorization. |
| FR3 | The system shall predict a label using a trained Logistic Regression model. |
| FR4 | The system shall calculate prediction probabilities for all possible labels. |
| FR5 | The system shall extract top keywords contributing to the prediction using model coefficients. |
| FR6 | The system shall generate a downloadable PDF report including input text, predicted label, keyword contributions, and class probabilities. |
| FR7 | The system shall allow users to download the trained model (.pkl) and vectorizer (.pkl). |

# 3. Non-Functional Requirements

|  |  |
| --- | --- |
| ID | Requirement Description |
| NFR1 | The system should respond to user input within 3 seconds. |
| NFR2 | The system should be able to handle up to 100 simultaneous requests. |
| NFR3 | The system must be portable and executable on any Python-supported environment (e.g., Google Colab). |
| NFR4 | The system must validate and sanitize user inputs to prevent crashes. |

# 4. System Components

|  |  |
| --- | --- |
| Component | Description |
| Vectorizer | Converts raw text into TF-IDF feature vectors. |
| Model | Logistic Regression model that outputs class predictions and probabilities. |
| PDF Generator | Creates a downloadable PDF report using `fpdf`. |
| Storage | Saves and loads models using `joblib`. |

# 5. User Interaction Flow

1. 1. User enters a symptom text.
2. 2. System vectorizes the text.
3. 3. Model predicts the class and calculates probabilities.
4. 4. System extracts top keywords.
5. 5. A PDF report is generated and offered for download.
6. 6. Optionally, user can download the model and vectorizer files.

# 6. Error Handling Requirements

|  |  |  |
| --- | --- | --- |
| ID | Error Condition | Expected Behavior |
| ER1 | Input is empty | Return a warning: 'Input text cannot be empty.' |
| ER2 | Text has no recognizable features | Return default classification with low confidence. |
| ER3 | Model/vectorizer not found | Return error: 'Model files are missing.' |