A Project Report on

"Zero-Shot Disease Prediction System Using Natural Language Processing"

Submitted to

DR. BABASAHEB AMBEDKAR TECHNOLOGICAL UNIVERSITY, LONERE

in partial fulfillment of the requirement for the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE & ENGINEERING
By

Pruthviraj Shyamrao Tarode
Vedant Naresh Karodkar
Mayur Ukandji Narwade
Shubham Kunturwar

Under the Guidance
of

Ms. Nitu L.Pariyal

(Department of Computer Science and Engineering)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
MAHATMA GANDHI MISSION'S COLLEGE OF ENGINEERING
NANDED (M.S.)

Academic Year 2025-26

Certificate

This is to certify that the project entitled

“Zero-Shot Disease Prediction System Using Natural Language Processing”

being submitted by ***Mr. Pruthviraj Shyamrao Tarode, Mr. Vedant Naresh Karodkar, Mr. Mayur Ukandji Narwade , Mr. Shubham Kunturwar*** *to the Dr. Babasaheb Ambedkar Technological University, Lonere , for the award of the degree of Bachelor of Technology in Computer Science and Engineering, is a record of bonafide work carried out by them under my supervision and guidance. The matter contained in this report has not been submitted to any other university or institute for the award of any degree.*

Ms. Nitu L.Pariyal

Project Guide

Dr. A. M. Rajurkar
H.O.D
Computer Science & Engineering

Dr. G. S. Lathkar
Director
MGM’s College of Engg., Nanded

ACKNOWLEDGEMENT

We are greatly indebted to our project guide, **Ms. Nitu L.Pariyal** , for her able guidance, and we would like to thank her for her help, suggestions, and numerous helpful discussions.

We gladly take this opportunity to thank **Dr. A. M. Rajurkar** (Head of Computer Science and Engineering, MGM's College of Engineering, Nanded).

We are heartily thankful to **Dr. G. S. Lathkar** (Director, MGM's College of Engineering, Nanded) for providing facilities during the progress of the project and for her kind guidance and inspiration.

Last but not least, we are also thankful to all those who helped directly or indirectly in the complete and successful development of this project.

With Deep Reverence,

Pruthviraj Shyamrao Tarode_102

Vedant Naresh Karodkar_170

Mayur Ukandji Narwade_177

Shubham Kunturwar_175

[B. Tech-CSE-A]

ABSTRACT

The Zero-Shot Disease Prediction System is an AI-driven healthcare solution designed to interpret natural-language symptom descriptions and generate medically meaningful predictions without requiring disease-specific training data. The system integrates advanced NLP technologies—including transformer-based multilingual embeddings, zero-shot learning, FAISS similarity search, and Natural Language Inference (NLI)—to evaluate the semantic relevance and logical consistency of user-provided symptoms. By supporting English, Hindi, Marathi, and mixed-language inputs, the system ensures accessibility for a diverse user population and adapts effectively to informal, conversational symptom descriptions.

The prediction workflow converts user input into semantic embeddings, retrieves the closest medical concepts, validates them through NLI reasoning, and ranks the results based on clinical relevance. A rule-based triage module further enhances user guidance by categorizing symptoms into low, medium, or high urgency levels, encouraging timely medical decision-making. To improve user experience, a responsive web interface was developed with HTML, CSS, and JavaScript, allowing seamless interaction with a FastAPI backend responsible for preprocessing, embedding generation, retrieval, validation, and response formatting.

Comprehensive experimental evaluations demonstrate that the system performs efficiently, delivering accurate predictions within seconds while maintaining robustness across multilingual and unstructured inputs. Additionally, the inclusion of features such as voice input, an intuitive welcome page, and a medical triage history form enriches usability and aligns the system with real-world healthcare needs.

Overall, this project showcases the potential of zero-shot learning and multilingual NLP in medical applications, offering a scalable, flexible, and user-centric approach for early disease awareness and preliminary clinical support.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

The rapid evolution of artificial intelligence in recent years has reshaped the way digital systems understand and interact with human language. Modern AI models have progressed far beyond basic keyword matching and now possess the ability to grasp context, meaning, tone, and subtle linguistic variations. This development has had a profound impact on fields where human communication is diverse and unpredictable, and healthcare is undoubtedly one of the most prominent among them. When individuals communicate their symptoms, they do so in highly personal ways. Two people with the same condition might describe it completely differently depending on their vocabulary, emotional state, native language, and cultural background. Such natural variation creates a challenge for traditional medical prediction systems, which typically depend on structured datasets with predefined symptom labels. These systems often require users to select symptoms from menus or enter information in specific formats and fail to interpret free-text expressions that reflect how people actually talk about their health.

In many real-world scenarios, constructing comprehensive disease-specific training datasets is impractical. Rare conditions, newly emerging diseases, and region-specific illnesses often lack sufficient annotated data. Furthermore, uncertainty in symptom presentation adds to the complexity, as no two patients experience or describe illnesses in the same way. In this context, zero-shot learning introduces a powerful alternative by enabling a model to infer disease predictions from the meaning of symptom descriptions rather than relying solely on labeled examples. A zero-shot disease prediction system analyzes the semantic content of a user’s input and compares it to disease descriptions, allowing it to identify relevant conditions even without having been explicitly trained on them. This is especially transformative for healthcare applications, where early guidance can help individuals recognize when a symptom should not be ignored. In regions with limited access to medical infrastructure or where people hesitate to seek immediate clinical help, such a system can act as a supportive tool that encourages informed decision-making.

1.1 Background of the Study

The incorporation of technology into healthcare has been ongoing for decades, yet early systems were largely constrained by rigid structures and limited linguistic understanding. They required users to enter symptoms in standardized formats or select from predefined lists, which ignored the natural variability of human expression. These systems could not interpret free-text inputs such as "I get breathless when walking upstairs" or regionally expressed sentences like "मुझे चक्कर और कमजोरी महसूस हो रही है," because they lacked the capacity to analyze meaning beyond fixed rules. This created a persistent gap between how people communicate in everyday life and how machines interpret information.

As digital communication expanded globally, people increasingly used the internet to describe and seek explanations for their symptoms. Their language was free, unstructured, and highly varied, incorporating colloquial phrases, mixed languages, and informal expressions. This shift highlighted the need for healthcare technologies that could understand natural human communication rather than enforce rigid templates. The emergence of deep learning and natural language processing marked the turning point. Transformer-based models, multilingual embeddings, and context-aware representations enabled machines to interpret sentences at a semantic level rather than a structural one. Zero-shot learning grew out of these advancements, providing a method for predicting unseen classes based solely on descriptive knowledge.

This approach aligns closely with the realities of healthcare, where collecting large annotated datasets for every disease is unrealistic. It also addresses the linguistic diversity in countries like India, where individuals often switch between languages within a single sentence or express medical concerns in their native tongue. With the help of multilingual transformer models, systems can now understand such natural expressions without requiring language-specific training. The background of this study is therefore grounded in the intersection of language, technology, accessibility, and the growing need for intelligent systems capable of understanding people as they naturally speak.

1.2 Problem Statement

The central issue addressed in this project is the inability of traditional medical prediction systems to understand free-text symptom descriptions across different languages and expressions. Most existing digital tools still depend on structured inputs, standard symptom lists, or training datasets that only represent a limited range of diseases. Such systems fail to recognize symptoms described in informal language or mixed linguistic formats, which are common in everyday communication. This becomes especially problematic when users describe vague or overlapping symptoms or when diseases present themselves differently in different individuals.

Furthermore, the lack of accessible and timely medical guidance exacerbates this problem. Many people avoid visiting doctors due to distance, financial limitations, fear, or underestimation of their symptoms. Without a reliable preliminary guidance system, individuals may delay seeking help until conditions worsen. In addition, it is impossible to create exhaustive training datasets for every disease, particularly rare conditions or rapidly evolving illnesses. The linguistic diversity in multilingual societies further complicates the situation, as models trained on one language often fail to interpret symptoms phrased in another.

Thus, the problem addressed by this project lies in designing a system capable of understanding natural symptom descriptions regardless of phrasing, language, or completeness, and able to predict possible diseases without relying on disease-specific training data. This requires a more flexible, adaptive, and semantically driven approach to medical prediction.

1.3 Objective of the Project

The main objective of this project is to develop a zero-shot disease prediction system that can analyze natural-language symptom descriptions and identify likely medical conditions through semantic understanding rather than traditional training. The system aims to interpret the user's input by converting it into meaningful numerical representations using advanced transformer models. These embeddings capture contextual information and represent the underlying meaning of the symptoms.

Once embedded, the system compares the user’s input with a database of disease descriptions using FAISS, a highly efficient similarity search engine. This allows the system to retrieve diseases that resemble the meaning of the input, even without having been trained on them. To ensure logical coherence, a Natural Language Inference model evaluates whether the user’s symptoms align with each disease description. This makes predictions medically meaningful rather than merely statistically similar.

The project also aims to support multiple languages, enabling users to describe symptoms in English, Hindi, Marathi, or other supported languages. It focuses on creating a system that does not require medical knowledge from the user and guides them naturally. Beyond identifying diseases, the system aims to provide suggestions related to diagnostic tests, specialists, and severity levels through triage logic. The overall objective is to deliver an accessible and intelligent health-support tool that can guide users toward timely medical consultation.

1.4 Significance of the Study

The significance of this study lies in its capacity to address real-world challenges faced by individuals seeking preliminary medical guidance, especially in environments where healthcare support is limited or difficult to access. From a technological standpoint, the system highlights how zero-shot learning and advanced natural language processing techniques can be effectively utilized to overcome the limitations of traditional machine learning models. By removing the dependence on disease-specific training datasets, the model becomes inherently more flexible, scalable, and sustainable in the long term. Its ability to interpret semantic meaning allows it to adapt effortlessly to new diseases, updated medical descriptions, and diverse linguistic patterns, making it future-ready in a constantly evolving healthcare landscape.

From a societal perspective, the system promotes accessible and early-stage healthcare awareness. Many individuals delay consulting medical professionals because of financial constraints, fear, stigma, or geographical barriers. A zero-shot disease prediction system provides an initial understanding of symptoms based solely on how users naturally describe their condition, encouraging timely and informed decisions.

This can potentially reduce complications caused by delayed diagnosis and empower individuals to take control of their health.

The multilingual capability of the system adds substantial value to its relevance. By allowing users to express symptoms in their native or mixed languages, the model becomes more approachable and culturally sensitive. This inclusivity not only strengthens user trust but also bridges the communication gap for populations that may struggle with English or formal medical terminology. It ensures that healthcare technology remains accessible to people across different socio-linguistic backgrounds.

Moreover, the study contributes academically by demonstrating a structured and practical integration of zero-shot learning, semantic embeddings, similarity search mechanisms, and logical inference models. It showcases how these emerging technologies can be harmonized to build real-world healthcare solutions that are both effective and user-friendly. Beyond contributing to academic knowledge, the project opens pathways for future research in digital healthcare, multilingual AI, and human-centered medical support systems. It reflects how intelligent systems can complement clinical practice by providing accurate, fast, and meaningful preliminary insights to users worldwide.

1.5 Report Organization

• Chapter 1 – Introduction

This chapter introduces the motivation behind developing an AI-based medical assistant, outlining the challenges in traditional diagnostic systems and the need for a zero-shot, multilingual, and intelligent disease prediction model. It also defines the problem statement, objectives, scope, and significance of the proposed system.

• Chapter 2 – Literature Survey

Provides an in-depth review of existing research on machine learning, deep learning, zero-shot learning, FAISS similarity search, natural language inference, and multilingual NLP. It also includes the existing system diagram, identifies limitations of current approaches, and highlights how the proposed system improves upon earlier methods.

- **Chapter 3 – System Design**

Describes the overall architecture of the proposed system, including UML diagrams, system components, module interactions, data flow, security architecture, and database design. This chapter explains how each design decision contributes to building an efficient, scalable, and intelligent prediction framework.

- **Chapter 4 – Implementation Details**

Covers all practical implementation steps such as preprocessing, embedding generation, FAISS indexing, NLI reasoning, triage classification, and backend API development using FastAPI. It also explains frontend design, performance optimization, scalability considerations, and module-wise execution details.

- **Chapter 5 – Results and Discussion**

Presents real system outputs, prediction results, triage classification screenshots, NLI validation results, and performance analysis. This chapter evaluates the effectiveness of the system and discusses its real-world applicability, accuracy, strengths, and improvements.

- **Conclusion**

Summarizes the overall project, key findings, learning outcomes, and the long-term relevance of AI-driven medical assistance. It also reflects on the project's contribution to healthcare technology and outlines possible improvements for future versions.

Chapter 2

LITERATURE REVIEW

The evolution of intelligent medical prediction systems is a product of decades of research in artificial intelligence, clinical informatics, computational linguistics, and human–computer interaction. Early diagnostic technologies were designed to assist healthcare professionals by providing computational support for decision-making. However, these systems were limited by rigid structures and narrow functionality. As technology advanced, researchers and developers began exploring more dynamic approaches that could adapt to the complexities of real-world medical data. Modern systems incorporate deep learning, semantic embeddings, multilingual natural language understanding, and zero-shot learning to interpret unstructured symptom descriptions expressed in diverse languages and styles. This chapter presents a comprehensive review of the theoretical, technological, and research developments that paved the way for the zero-shot disease prediction system developed in this project.

2.1 Review of Existing Medical Prediction Platforms

Several existing medical diagnostic systems and AI-based symptom checkers have been developed in recent years. These platforms attempt to assist users by interpreting symptoms and providing possible medical conditions. While they differ in functionality and complexity, they provide an important foundation for understanding how automated diagnostic tools operate and where improvements are required. A few notable existing systems and similar projects are discussed below.

(a) WebMD Symptom Checker

WebMD is one of the most widely used online symptom-checking platforms. Users manually select symptoms from predefined lists, and the system generates possible conditions using rule-based and statistical medical data. However, WebMD cannot interpret free-text natural language inputs, and its predictions are restricted to the conditions stored in its symptom-disease mapping database. It also lacks multilingual support and cannot perform logical reasoning using AI models as shown in Fig 2.1. ReferenceLink: https://symptoms.webmd.com

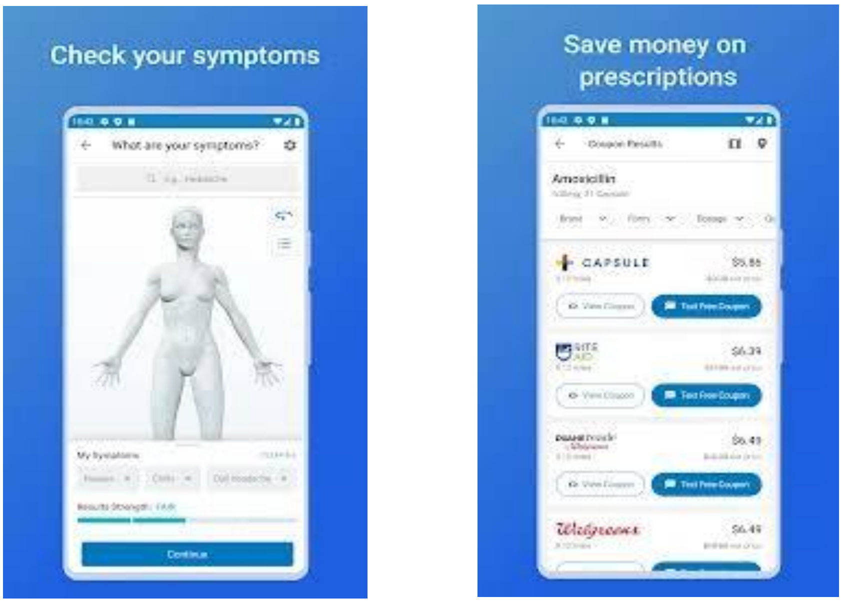


Fig 2.1 WebMD Symptom Checker

(b) Ada Health – AI Medical Assessment App

Ada Health uses a machine learning–based approach to analyze user symptoms and generate health assessments. It provides a chatbot-style interface where users answer structured questions. While Ada uses advanced AI models, it does not support open-ended natural language descriptions, nor does it follow a zero-shot learning approach. Its predictions are still tied to curated training datasets as shown in Fig 2.2.

Reference Link: https://ada.com

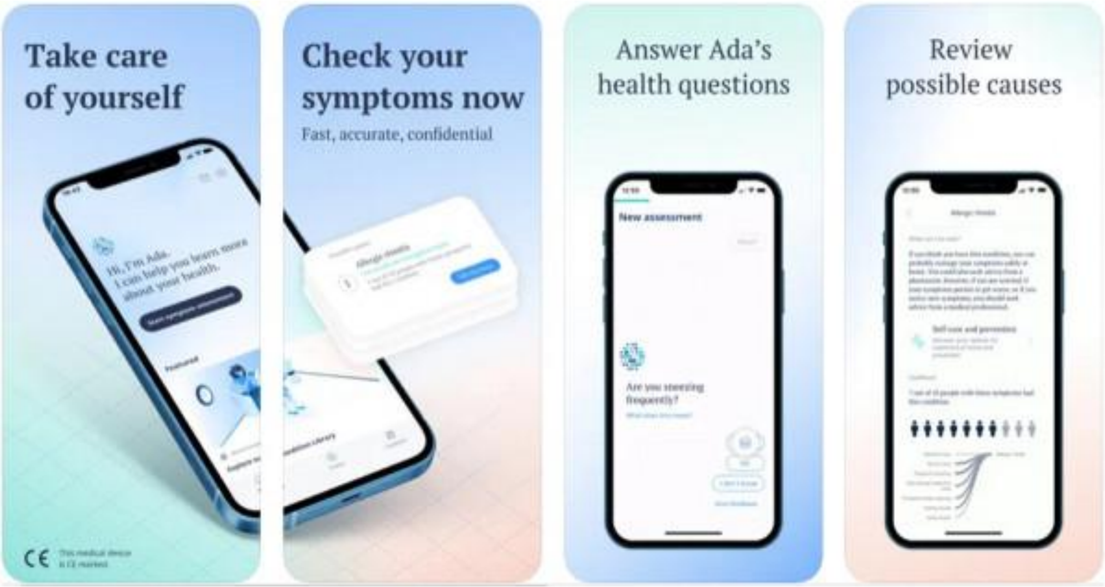
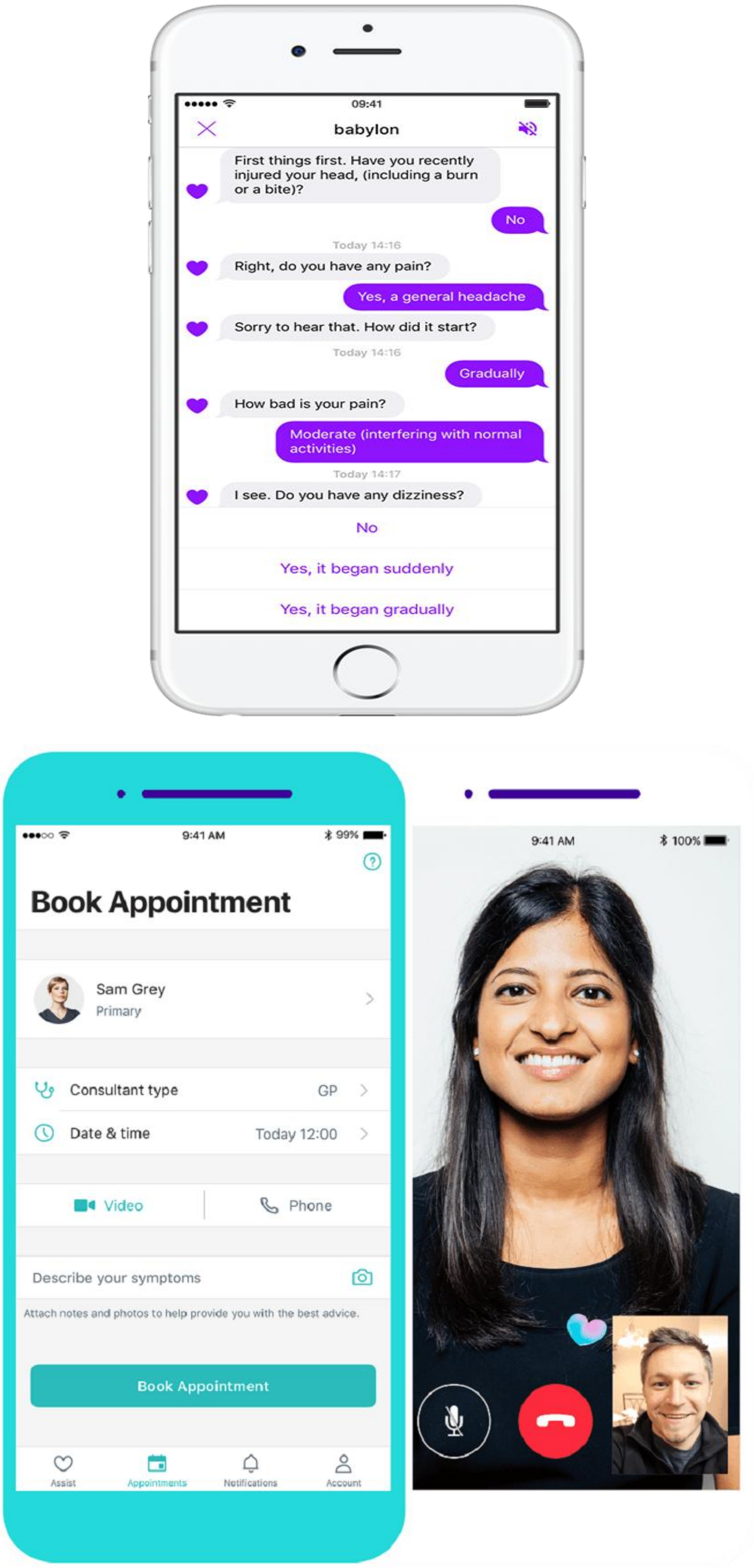

Take care of yourself
Hi, I'm Ada. I can help you learn more about your health.
Check your symptoms now
Fast, accurate, confidential
Answer Ada's health questions
New assessment
Are you sneezing frequently?
Review possible causes

Fig 2.2 Ada Health – AI Medical Assessment App

(c) Babylon Health – AI Consultation System

Babylon Health offers an AI-driven medical assistant that evaluates symptoms through conversational interaction. Although more advanced than traditional rule-based systems, its disease prediction accuracy depends heavily on large supervised training datasets, which limits its ability to adapt to new or rare medical conditions. The system follows a predefined question–answer flow, reducing flexibility when users describe symptoms freely in natural language. It does not use FAISS vector search or NLI validation, meaning it cannot perform semantic similarity reasoning or logical consistency checks between symptoms and predicted diseases. As a result, adding new medical conditions requires retraining the model, which affects scalability, as shown Fig2.3.

Reference Link: https://www.babylonhealth.com



First things first. Have you recently injured your head, (including a burn or a bite)?

No

Today 14:16

Right, do you have any pain?

Yes, a general headache

Sorry to hear that. How did it start?

Today 14:16

Gradually

How bad is your pain?

Moderate (interfering with normal activities)

Today 14:17

I see. Do you have any dizziness?

No

Yes, it began suddenly

Yes, it began gradually

Sam Grey
Primary

Consultant type GP

Date & time Today 12:00

Describe your symptoms

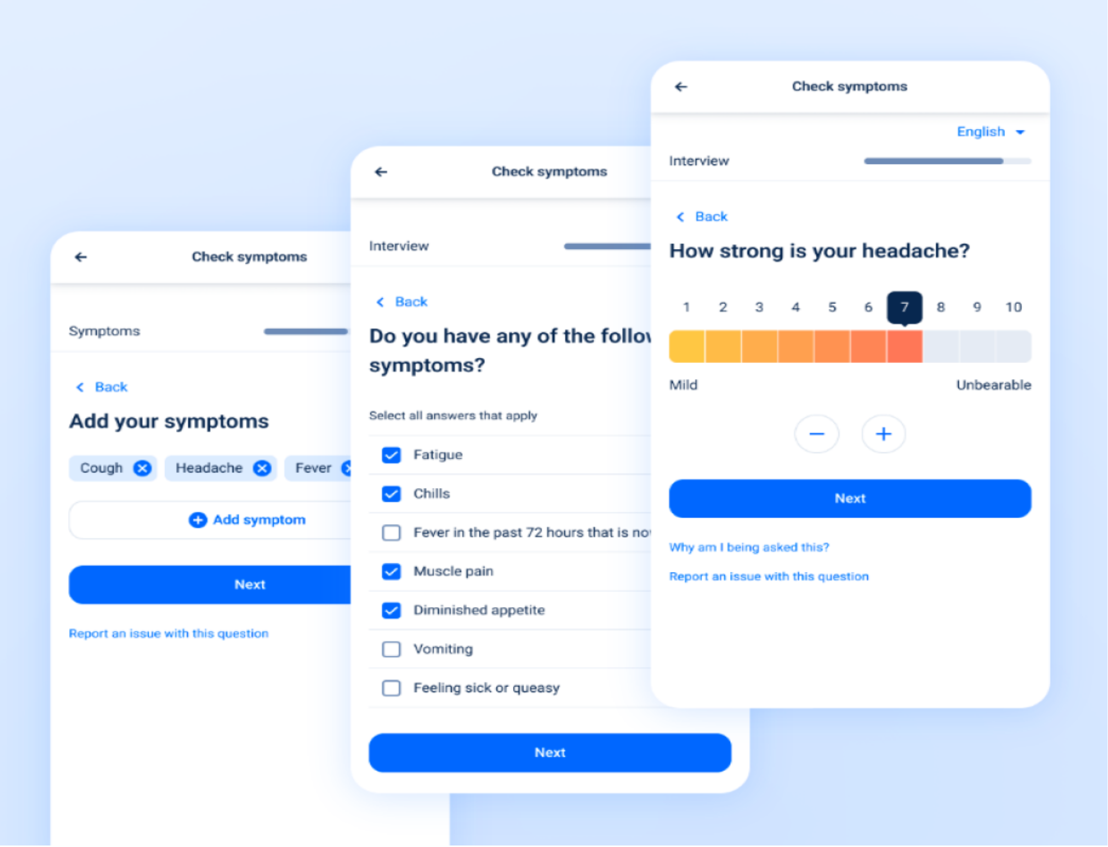
Attach notes and photos to help provide you with the best advice.

Fig 2.3 Babylon Health – AI Consultation System

(d) Infermedica – Symptom Triage Engine

Infermedica provides a medical inference engine that performs symptom assessment and triage. It uses probabilistic reasoning rather than zero-shot learning or semantic embedding techniques. The system cannot interpret multilingual free-text input, making it less flexible compared to modern NLP-based approaches as shown in Fig 2.4.

Reference Link: https://infermedica.com



Add your symptoms

Do you have any of the follow
symptoms?

Select all answers that apply

☑ Fatigue
☑ Chills
□ Fever in the past 72 hours that is no
☑ Muscle pain
☑ Diminished appetite
□ Vomiting
□ Feeling sick or queasy

How strong is your headache?

Fig 2.4 Infermedica – Symptom Triage Engine

2.2 Evolution of Medical Diagnostic Systems

The earliest generation of computerized diagnostic tools emerged in the 1970s and 1980s, when expert systems were developed to emulate human decision-making. These systems, including landmark projects like MYCIN and INTERNIST-I, relied entirely on manually encoded rules that described medical knowledge in "if–then" form. These systems represented significant technological achievements for their time, yet they suffered from fundamental limitations. Their reliance on expert-created rules made them slow to update, difficult to scale, and unable to adapt to new diseases or nuanced expressions of symptoms.

As medical knowledge expanded and user expectations evolved, the limitations of rule-based systems became increasingly apparent. These systems were incapable of handling ambiguity, linguistic variation, or free-text input. They failed when faced with expressions such as “my chest feels heavy at night” or “मुझे हल्का चक्कर आ रहा है,” because they lacked the ability to understand natural human language. The inability to interpret subjective experiences or informal descriptions highlighted the need for more flexible, data-driven systems capable of learning from real-world patterns rather than static rules.

2.3 Transition to Traditional Machine Learning

The introduction of traditional machine learning marked a major shift in medical prediction research. Models such as logistic regression, decision trees, support vector machines, and ensemble methods brought statistical pattern recognition into healthcare. These algorithms could analyze structured datasets composed of symptoms, demographics, and diagnostic outcomes, and learn associations automatically.

However, despite their success in structured-data environments, traditional machine learning approaches still faced foundational challenges. The most significant limitation was their dependency on large labeled datasets, which are often scarce in healthcare due to privacy concerns, variability in clinical documentation, and the rarity of certain diseases. These models also struggled with natural language content because they relied on numerical inputs rather than textual information. Techniques such as bag-of-words or TF–IDF vectors were used to convert text into numerical form, yet these early NLP methods failed to capture the deeper meaning, context, and relationships between words. They considered “chest pain” and “pain in chest” as unrelated phrases, even though medically they represent the same symptom.

This inability to understand linguistic structure and semantic nuances restricted the effectiveness of traditional machine learning in real-world diagnostic scenarios. Thus, the search for more sophisticated language understanding techniques intensified.

2.4 Rise of Deep Learning and Linguistic Modeling

Deep learning introduced a revolutionary shift in the ability of machines to process unstructured textual information. Recurrent neural networks (RNNs) and Long Short-Term Memory (LSTM) models improved the interpretation of sequential data by incorporating memory mechanisms that captured context across multiple words. This advancement allowed healthcare models to process longer descriptions and detect patterns indicative of certain diseases.

Even with these improvements, RNNs struggled with long-range dependencies and multilingual complexity. The introduction of the transformer architecture fundamentally changed natural language processing. Transformational models such as BERT (Bidirectional Encoder Representations from Transformers), GPT, RoBERTa, and XLM-RoBERTa enabled machines to process text bidirectionally and understand the relationships between every pair of words in a sentence. This led to deeper comprehension of natural-language symptom descriptions.

Transformers also introduced the concept of embeddings—dense numerical representations capturing meaning, context, and linguistic patterns. Unlike earlier vectorization techniques, embeddings allow models to understand that “shortness of breath,” “difficulty breathing,” and “breathlessness” refer to related medical concepts. This new level of linguistic intelligence laid the foundation for more advanced diagnostic systems, particularly those based on semantic understanding.

2.5 Zero-Shot Learning and Its Importance in Medical AI

Zero-shot learning (ZSL) emerged as an innovative solution to one of the most persistent challenges in healthcare AI: the scarcity of labeled training data for numerous diseases. ZSL allows a model to recognize categories it has never seen during training by leveraging the semantic descriptions of those categories. Instead of learning from examples, the system learns from textual definitions or conceptual descriptions.

In the medical domain, this approach is especially valuable because diseases often lack substantial datasets due to rarity, novelty, or privacy concerns. Moreover, new diseases can emerge at any time, as evidenced during the COVID-19 pandemic, when early detection tools were needed before large datasets existed. Zero-shot learning bypasses the need for disease-specific training and instead relies on the model's linguistic understanding of disease descriptions.

ZSL aligns perfectly with the nature of symptom interpretation, where human expression is diverse, unpredictable, and varied across cultures. By relying on semantic reasoning rather than memorized patterns, zero-shot models can interpret free-text symptoms regardless of phrasing or language, making them highly inclusive and adaptive.

2.6 Semantic Similarity Search and the Role of FAISS

Embeddings enable meaningful representation of symptoms and diseases, but these representations must be compared efficiently to identify the closest matches. This requirement led to the adoption of FAISS (Facebook AI Similarity Search), a high-performance library designed to search through millions of vectors in milliseconds.

FAISS plays a critical role in enabling real-time disease prediction. When a user enters symptoms, the system converts them into embeddings and compares them with pre-computed disease embeddings using FAISS. The library uses advanced indexing structures, clustering techniques, and GPU acceleration to identify the nearest neighbors quickly and accurately. Without FAISS, performing such searches manually would be computationally expensive and too slow for practical use.

In the context of zero-shot disease prediction, FAISS serves as the engine that retrieves the most semantically relevant disease candidates, forming the basis for further reasoning and validation.

2.7 Logical Validation through Natural Language Inference

Similarity-based retrieval alone cannot guarantee medical correctness. Two texts may appear similar at the linguistic level but differ significantly in meaning. For example, “I do not have chest pain” and “I have chest pain” share similar vocabulary but communicate opposite medical states. To resolve such ambiguities, Natural Language Inference (NLI) models are used.

NLI evaluates the logical relationship between two pieces of text—typically classifying the relationship as entailment, contradiction, or neutrality. In medical prediction systems, NLI ensures that the user’s symptoms logically support the potential disease. This reasoning layer adds a level of verification that prevents incorrect or misleading predictions. It acts as a safety mechanism, ensuring that the final output aligns with medical semantics rather than surface-level text similarity.

By integrating NLI, modern prediction systems achieve a balance between statistical relevance and logical consistency, creating more trustworthy results.

2.8 Growth of Multilingual NLP for Healthcare

Language diversity poses a major challenge in digital healthcare. In multilingual countries like India, users commonly describe symptoms in local languages or mixed-language formats. Traditional NLP systems designed primarily for English fail to interpret these inputs accurately.

Modern multilingual transformer models, such as mBERT, XLM-RoBERTa, and multilingual Sentence-BERT, solve this problem by generating unified embedding spaces for multiple languages. These models understand that “fever,” “bukhaar,” and “ताप” represent the same medical concept. This capability allows healthcare systems to become inclusive and accessible to broader populations.

Multilingual NLP research emphasizes the importance of bridging language gaps in healthcare technology. By supporting diverse expressions, these systems build trust and allow users to communicate naturally without adjusting to technical formats.

2.9 Consolidated Insights from Literature

The literature reviewed in this chapter demonstrates a clear trajectory of innovation in medical prediction systems. Early rule-based models provided structure but lacked adaptability. Traditional machine learning improved pattern recognition but struggled with free-text input and required large datasets. Deep learning introduced powerful models capable of contextual understanding, while transformers revolutionized natural language processing by capturing semantic meaning at an unprecedented level.

Zero-shot learning emerged as a groundbreaking paradigm capable of predicting unseen diseases through semantic reasoning rather than labeled data. FAISS added computational efficiency to large-scale embedding comparisons, and NLI contributed logical validation to ensure medically meaningful predictions. Multilingual NLP extended accessibility by enabling systems to understand symptoms expressed in different languages.

Collectively, these advancements created a strong theoretical foundation for developing intelligent, multilingual, and semantically aware disease prediction systems. The zero-shot disease predictor built in this project integrates these ideas into a unified architecture capable of handling real-world symptom descriptions with significant accuracy and inclusivity.

In addition, the reviewed literature highlights the growing importance of real-time performance, interpretability, and privacy in healthcare AI applications. It also emphasizes the need for systems that can operate effectively in low-resource settings where labeled medical data is scarce. The convergence of semantic intelligence with scalable computation is proving to be a key direction for next-generation clinical decision support tools. Furthermore, the integration of multilingual processing ensures that such systems can bridge healthcare access gaps across diverse populations.

Chapter 3

SYSTEM DESIGN

The system design phase forms a crucial part of the project as it defines how the proposed Zero-Shot Learning based Medical Assistant System will operate in a real-world environment. This phase focuses on translating the conceptual idea of the project into a well-structured technical framework that clearly explains the interaction between different components of the system. A carefully planned design ensures that the system functions efficiently, remains secure, and delivers accurate medical predictions to users.

The design of the system emphasizes the development of an intelligent, fast, and scalable medical assistance platform that is capable of interpreting natural language symptom descriptions and generating meaningful disease predictions without depending on disease-specific training data. To achieve this objective, the system follows a layered and modular architectural approach where individual modules such as input processing, semantic embedding generation, similarity search, logical validation, triage classification, and result presentation work in coordination with each other.

Both the structural and behavioral aspects of the system are explained using various UML and architectural diagrams such as the Component Diagram, Deployment Diagram, Activity Diagram, Data Flow Diagram, Use Case Diagram, Sequence Diagram, State Diagram, Security Architecture Diagram, and Entity Relationship Diagram. These diagrams help in visualizing system workflow, data movement, user interaction, internal state transitions, and applied security mechanisms.

Overall, this chapter provides a detailed blueprint of the system by describing how the different components are organized, how information flows across various stages of processing, and how the system maintains accuracy, performance, and security. This design foundation plays an important role in guiding the implementation and evaluation phases discussed in the following chapters. In addition, a well-defined system design helps in minimizing development errors, improving maintainability, and ensuring that future enhancements can be integrated smoothly without major structural changes. It also serves as a technical reference for developers and researchers who wish to understand, improve, adapt, or extend the system in the future.

3.1 Design Objectives

The primary objective of the system design is to develop a reliable, intelligent and user-friendly Zero-Shot Learning based Medical Assistant that can understand natural language symptom descriptions and generate accurate disease predictions without relying on disease-specific training datasets. The design aims to bridge the gap between complex medical information and ordinary users by allowing them to describe their health issues in simple, everyday language while still receiving structured and meaningful predictions.

Another major objective is to combine semantic understanding with logical validation so that the system does not depend only on keyword matching. The design ensures that the system first identifies semantically related diseases and then checks whether those diseases are logically supported by the symptoms. This layered reasoning approach improves the accuracy and trustworthiness of predictions.

The system is also designed to support fast response time so that users receive near real-time feedback, which is important in healthcare contexts. Security and privacy form a core part of the design objectives because the system deals with sensitive symptom information. Therefore, the design includes secure API communication, access control, and data protection mechanisms. In addition, the design aims for multilingual support so that the system can handle inputs in different languages, and for scalability so that new diseases, models and features can be integrated in future without disrupting the existing workflow.

3.2 Overall System Architecture

The overall system architecture of the Zero-Shot Medical Assistant follows a layered and modular structure. This architecture clearly separates user interaction, application logic, machine learning intelligence and data management so that each layer can be improved or replaced independently. The logical view of the system components and their interactions is depicted in Fig 3.1 Component Diagram – Software Architecture, while the physical deployment of these components across client devices, servers and external services is illustrated in Fig 3.2 Deployment Diagram – Infrastructure.

The frontend layer provides the user interface through which users enter symptoms, select the preferred language and view the prediction results. The API layer, built using FastAPI, receives requests from the frontend, validates inputs, applies security checks and forwards valid requests to the processing pipeline. The logic layer contains the core processing pipeline that controls input normalization, embedding generation, similarity search, NLI validation, ranking and triage. The machine learning layer includes the SentenceTransformer model, FAISS similarity search engine, XLM-RoBERTa NLI model and triage classifier, which together provide the actual intelligence of the system. The data layer stores disease descriptions, FAISS index files, user records, logs and configuration data.

As shown in Fig 3.2, the system is deployed in a way that allows it to handle multiple requests concurrently, balance load across servers and maintain high availability. This architecture ensures that the system remains scalable, maintainable and suitable for real-time medical assistance.

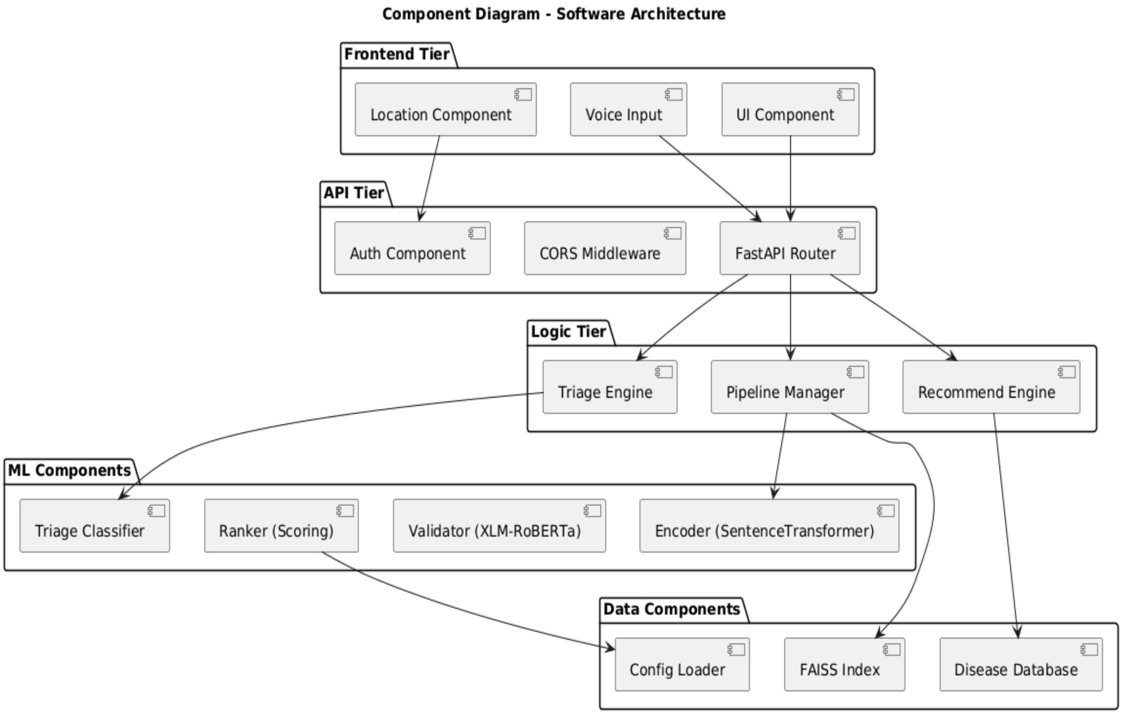


Fig 3.1 Component Diagram – Software Architecture

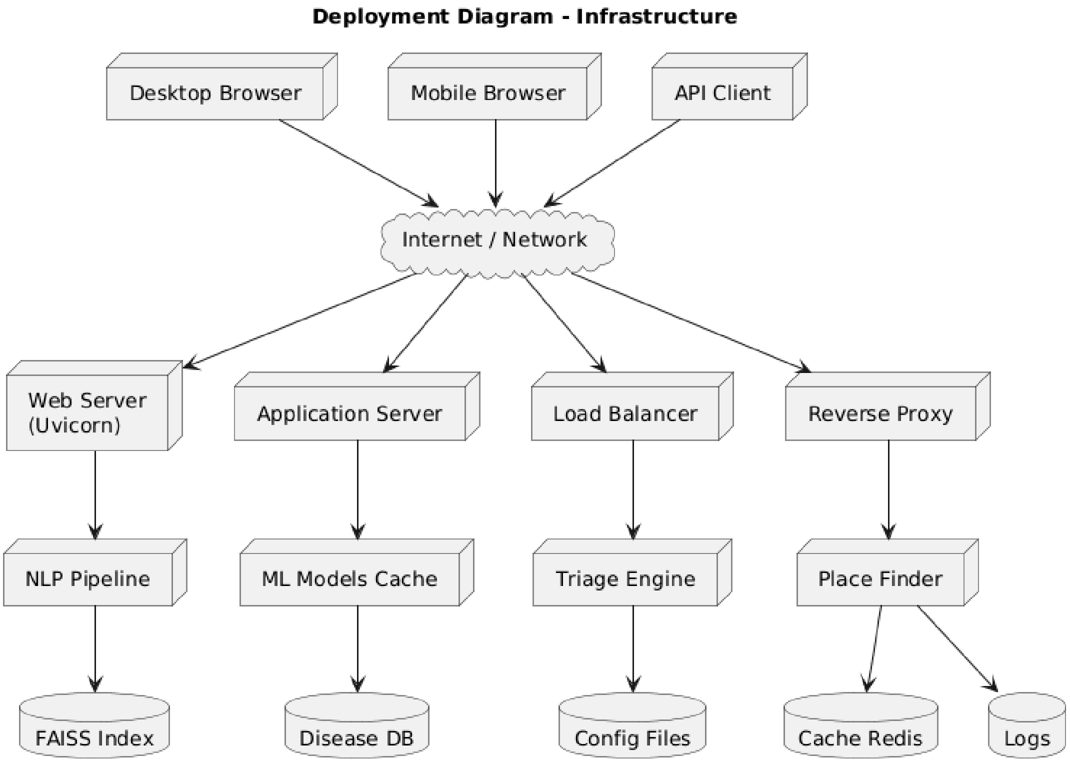


Fig 3.2 Deployment Diagram – Infrastructure

3.3 Input Processing and Normalization

User inputs are generally unstructured, informal and sometimes written in mixed languages. They may include spelling mistakes, abbreviations, emojis or incomplete phrases. Such raw text cannot be directly used by the embedding and similarity models. For this reason, the system includes an input processing and normalization module that prepares the text for further analysis. The overall flow of this phase is depicted in Fig 3.3 Activity Diagram – Symptom Analysis Workflow.

In this stage, unnecessary symbols, extra spaces and irrelevant punctuation are removed from the symptom description. The text is converted into a consistent format, such as lowercased form, to reduce variations caused by different writing styles. Simple spelling errors are corrected wherever possible and repeated words or noisy segments are normalized. When users provide symptoms in regional languages such as Hindi or Marathi, or in a mix of languages, the multilingual normalization ensures that the original medical meaning is preserved while still making the text suitable for the model.

By performing these steps, the normalization module improves the quality of the text that enters the semantic embedding stage. Clean and standardized input helps the remaining pipeline produce more accurate and stable results. Without proper normalization, the system would be highly sensitive to minor variations in user input.

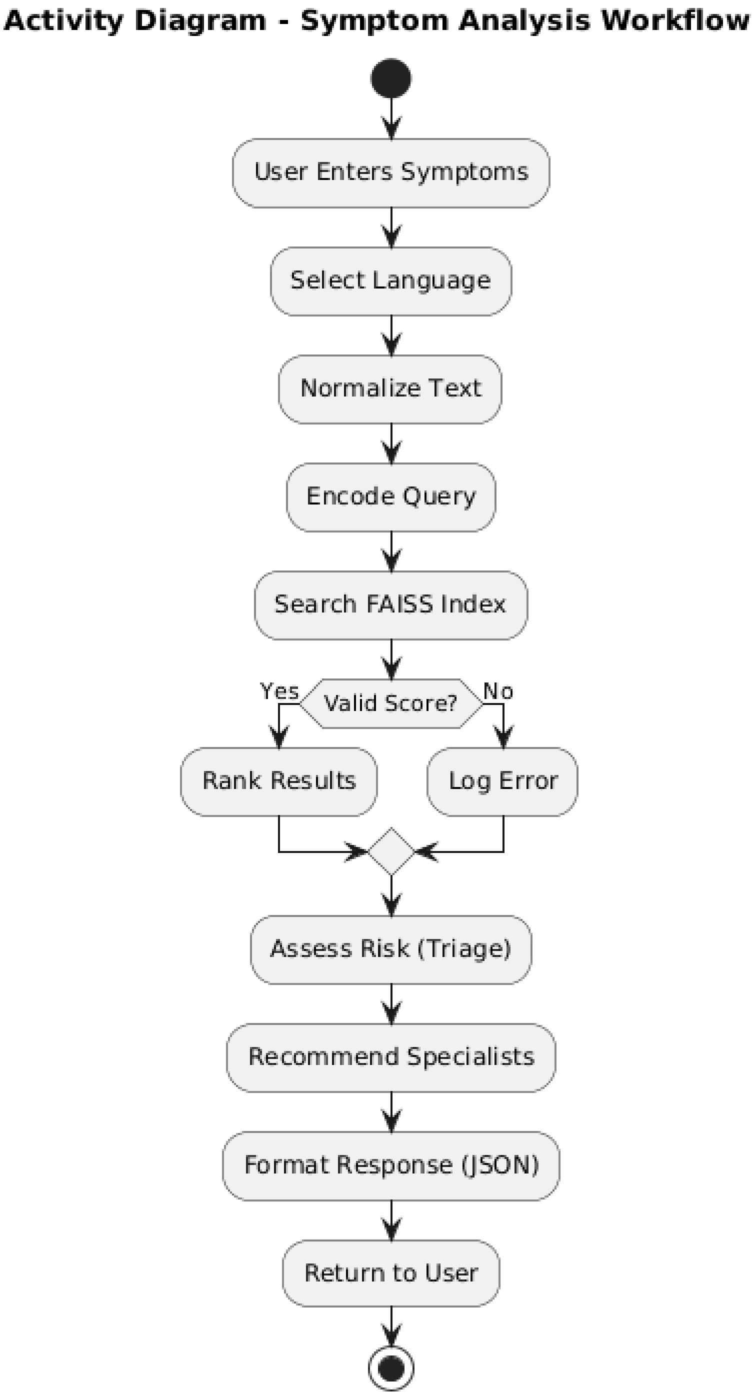


Fig 3.3 Activity Diagram – Symptom Analysis Workflow

3.4 Embedding Generation Using Sentence Transformers

Once the symptom description has been cleaned and normalized, it is passed to the embedding generation module. At this stage, the system uses a Sentence Transformer-based model to convert the input text into a high-dimensional numerical vector known as a semantic embedding. This vector captures the overall meaning of the sentence rather than just counting the individual words.

The goal of generating embeddings is to map semantically similar sentences close to each other in vector space. For example, phrases such as “tightness in chest,” “pressure in chest while breathing” and “feeling heaviness in chest” may be written differently, but medically they point towards related conditions. The embedding model is able to capture this similarity and represent these sentences in a way that allows effective comparison.

This semantic representation is particularly important for Zero-Shot Learning because the system is not trained specifically on each disease. Instead, it learns a general understanding of language and applies that understanding to match user inputs with disease descriptions. The complete transformation from text to vector lays the foundation for the similarity search performed in the next stage.

3.5 FAISS-Based Similarity Search

After the symptom text is converted into an embedding vector, the system needs to identify which diseases are most relevant to this embedding. For this purpose, a FAISS-based similarity search module is used. FAISS is a high-performance library designed for searching similar vectors in large collections, which makes it ideal for real-time applications.

The user’s embedding is compared against a large set of precomputed disease embeddings stored in a FAISS index. As shown in Figure 3.4 Data Flow Diagram – Symptom Processing Pipeline, the embedding is sent to the FAISS engine, which efficiently retrieves the top-k closest disease vectors based on distance metrics. These retrieved disease candidates represent the conditions that are semantically closest to the user’s symptom description.

This stage is optimized for speed so that even if the disease database grows larger, the system can still return results quickly. However, at this point, the diseases are selected mainly on semantic similarity and are not yet checked for logical consistency. Therefore, they are forwarded to the next validation step for further refinement.

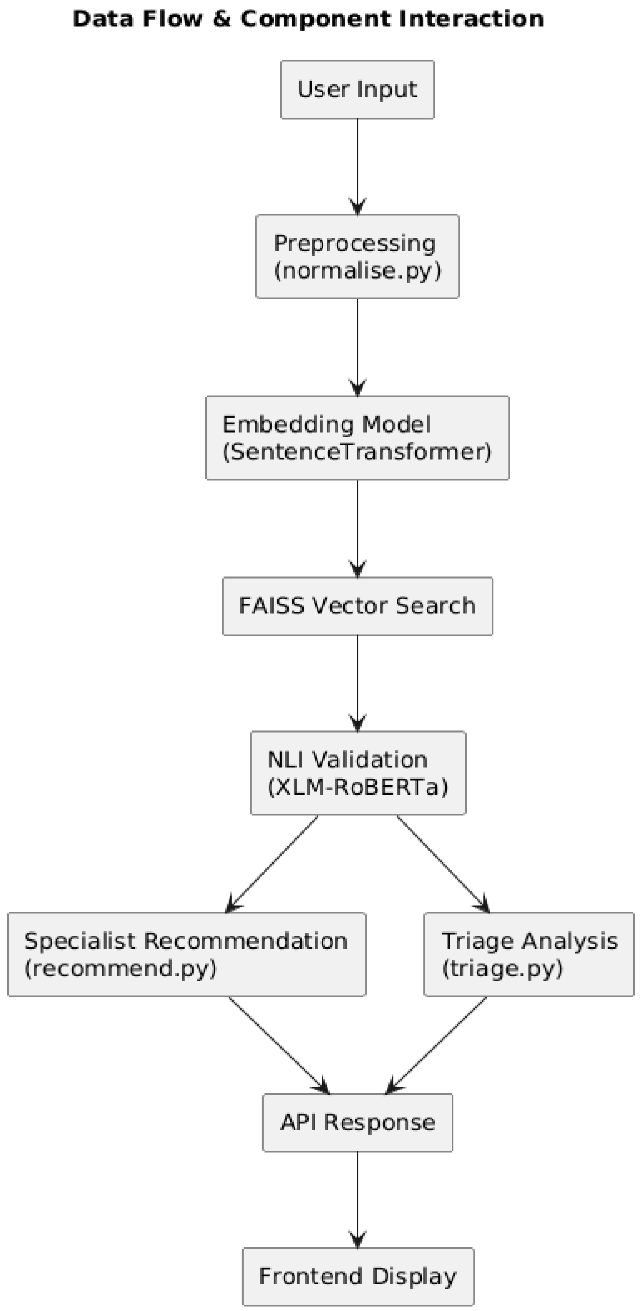

Data Flow & Component Interaction
User Input
Preprocessing (normalise.py)
Embedding Model (SentenceTransformer)
FAISS Vector Search
NLI Validation (XLM-RoBERTa)
Specialist Recommendation (recommend.py)
Triage Analysis (triage.py)
API Response
Frontend Display

Fig 3.4 Data Flow Diagram – Symptom Processing Pipeline

3.6 Natural Language Inference (NLI) Validation

Semantic similarity alone is not sufficient to ensure that a predicted disease truly matches the user’s symptoms. To add a layer of logical reasoning, the system uses a Natural Language Inference (NLI) validation module built using the XLM-RoBERTa model. In this stage, each candidate disease description retrieved from FAISS is paired with the user’s symptom description and passed through the NLI model.

The NLI model classifies the relationship between the symptom description (premise) and the disease description (hypothesis) as entailment, contradiction or neutral. Only those diseases that fall under the entailment class are considered logically supported by the symptoms. Contradicting or neutral relationships are filtered out. In this way, the NLI stage significantly reduces false positives and ensures that the final predictions are both semantically relevant and logically consistent with the symptoms reported by the user.

3.7 Disease Ranking

After the NLI validation is completed, the remaining set of disease predictions is further processed in the ranking stage. In this stage, each valid disease is scored using a combination of its semantic similarity distance and NLI confidence. Diseases with higher similarity and stronger entailment confidence are ranked higher in the final output.

The ranking module orders the diseases so that the most probable and medically relevant prediction appears at the top. The final ranked results are then formatted into a structured response that contains the disease name, confidence level, and other supporting information. This structured output is later used by the frontend to present the results clearly to the user.

3.8 Triage Classification

While disease prediction is important, it is equally critical to understand how serious the user’s condition might be. The triage classification module is responsible for assessing the urgency associated with the predicted diseases. It examines the symptoms for high-risk indicators such as chest pain, difficulty in breathing, high fever, sudden weakness or neurological issues.

Based on predefined medical rules and classification logic, the triage module assigns each case into one of three categories: low risk, medium risk or high risk. This classification helps users understand whether they need immediate medical attention, a prompt consultation or routine follow-up. By combining disease prediction with urgency assessment, the system becomes more useful in real-life healthcare situations.

3.9 Use Case Design

The functional behavior of the system from a user’s point of view is described through the use case design. The Use Case Diagram – ZSL Medical Assistant, shown in Fig 3.5 , presents the main interactions between the system and its actors. The primary actor is the patient, who can enter symptoms, choose a language, request disease prediction and view the triage result and recommendations. The doctor acts as a secondary actor who may view prediction summaries and use them as preliminary decision support. The admin oversees system configuration, data updates and monitoring.

This use case design helps to clearly define what operations are available to which type of user, and it also identifies system boundaries. It ensures that every feature implemented in the system has a clear purpose and a corresponding actor.

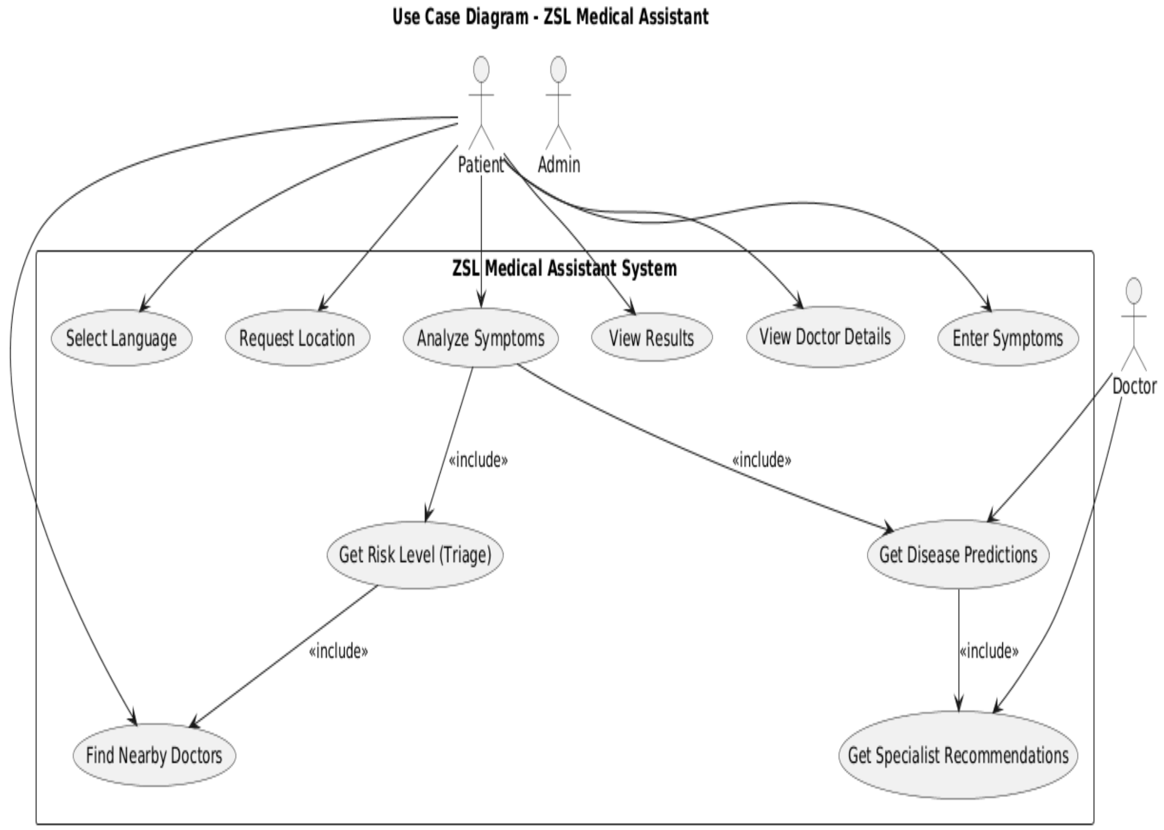


Fig 3.5 Use Case Diagram – ZSL Medical Assistant

3.10 Sequence and State Design

The detailed interaction between different components over time is explained through the Sequence Diagram – Symptom Analysis Flow, presented in Fig 3.6. This diagram shows how a user's request travels from the frontend to the FastAPI backend, moves through normalization, embedding, FAISS search, NLI validation, ranking, and triage, and then returns to the frontend as a final structured response. The sequence diagram helps in understanding the exact order of operations, the flow of data across modules, and how each processing step depends on the previous one. It provides a clear visualization of asynchronous communication and highlights the importance of parallel execution in improving system responsiveness. By illustrating the message-passing structure, the diagram assists developers in debugging, optimizing latency, and ensuring modular consistency across the pipeline.

The internal behavior of the system is further modeled using the State Diagram – Prediction Processing Lifecycle, shown in Fig 3.7. The system transitions through different states such as idle, receiving input, processing, validating, ranking, completed, and error. Each state represents a controlled phase in the lifecycle of a prediction request, ensuring that operations follow a predictable and stable flow. The state diagram clarifies how the system reacts to valid inputs, invalid inputs, exceptions, or timeouts, making error-handling strategies more transparent. It also ensures that the application maintains robustness by preventing undefined states and ensuring orderly recovery in case of failures. This structured state management contributes to the reliability and consistency of the prediction engine

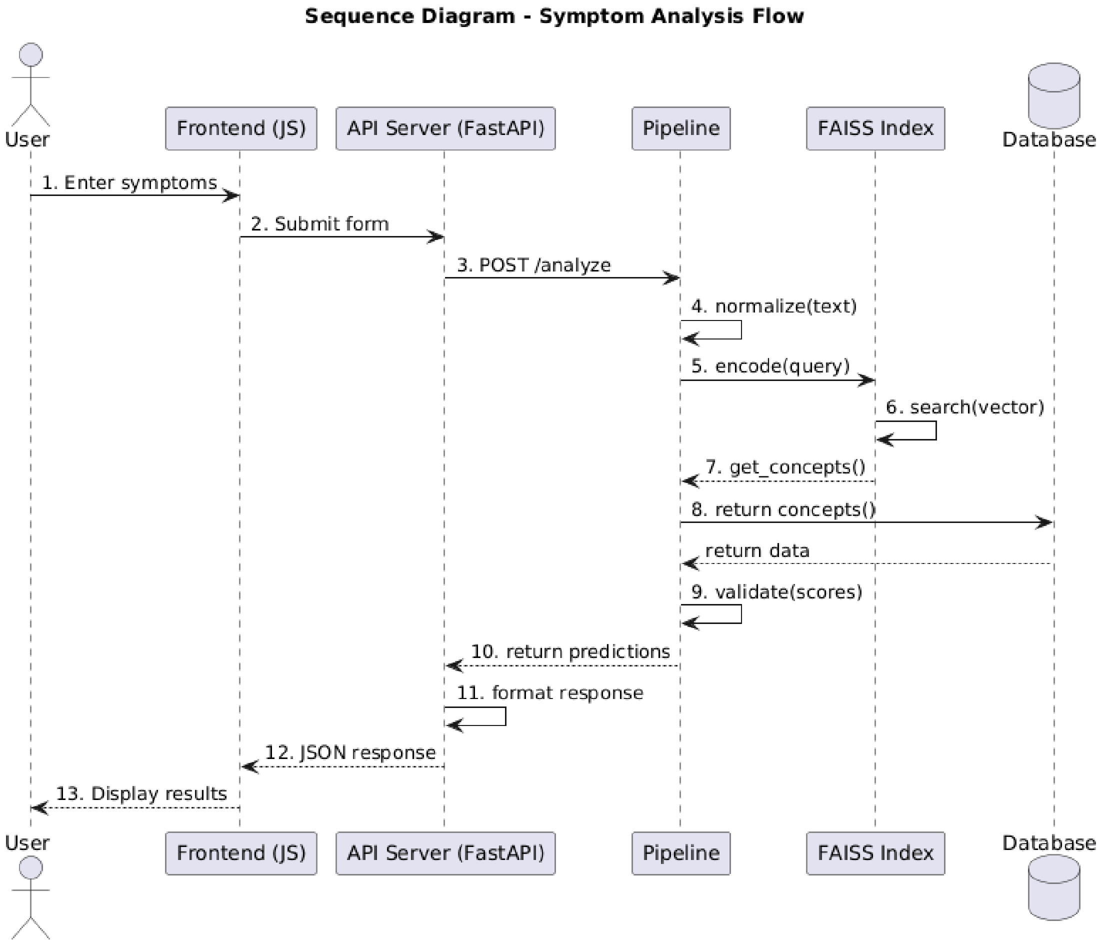


Fig 3.6 Sequence Diagram – Symptom Analysis Flow

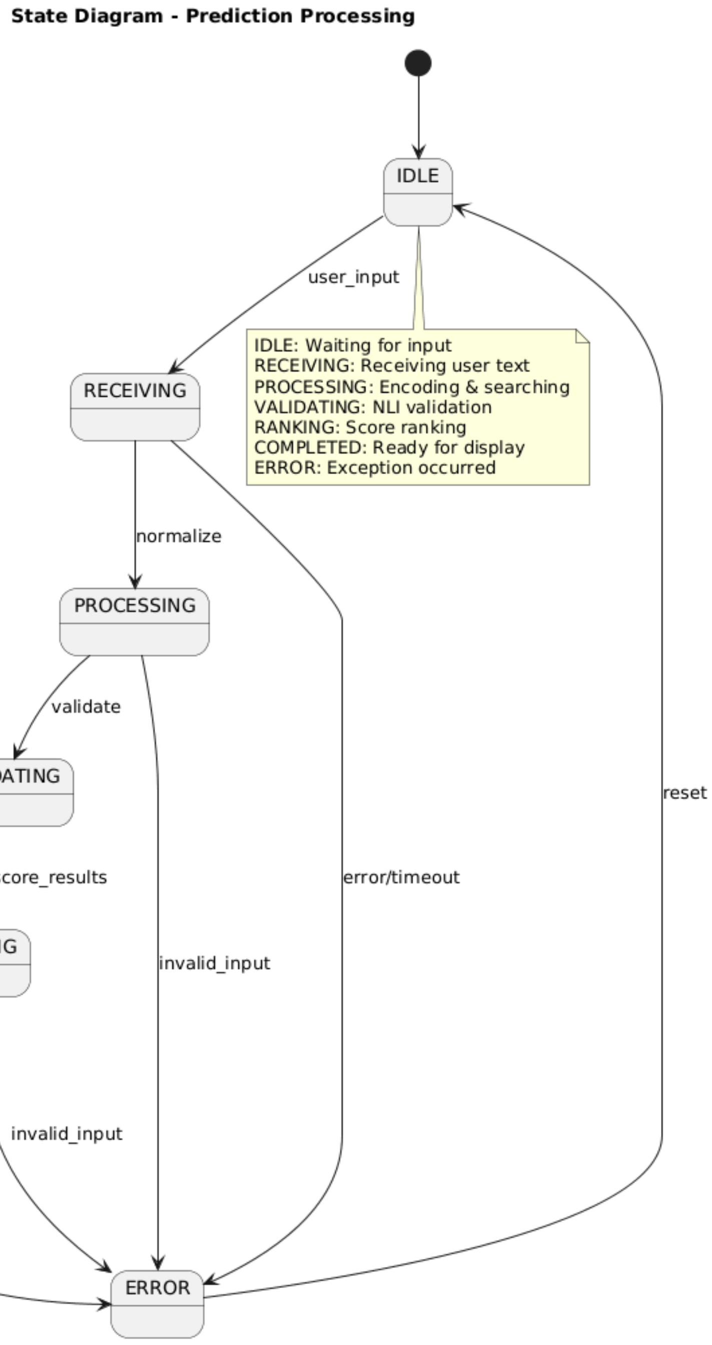


Fig 3.7 State Diagram – Prediction Processing

3.11 Security Architecture

Because the system handles sensitive healthcare-related information, security design is a critical aspect of the overall architecture. The Security Architecture Diagram, illustrated in Figure 3.8, shows the different layers of security applied to the system.

At the application layer, input validation and sanitization are used to prevent injection and cross-site scripting attacks. At the API layer, authentication, rate limiting and CORS policies help to control access and protect the service from abuse. At the data layer, encryption and secure storage ensure that sensitive information is not exposed in plain text. Infrastructure-level security is enforced using firewalls, HTTPS (SSL/TLS) communication and secure deployment practices. Logging and monitoring mechanisms are also included to detect unusual activities and to support auditing. Together, these measures create a multi-layered defense strategy to protect user privacy and system integrity.

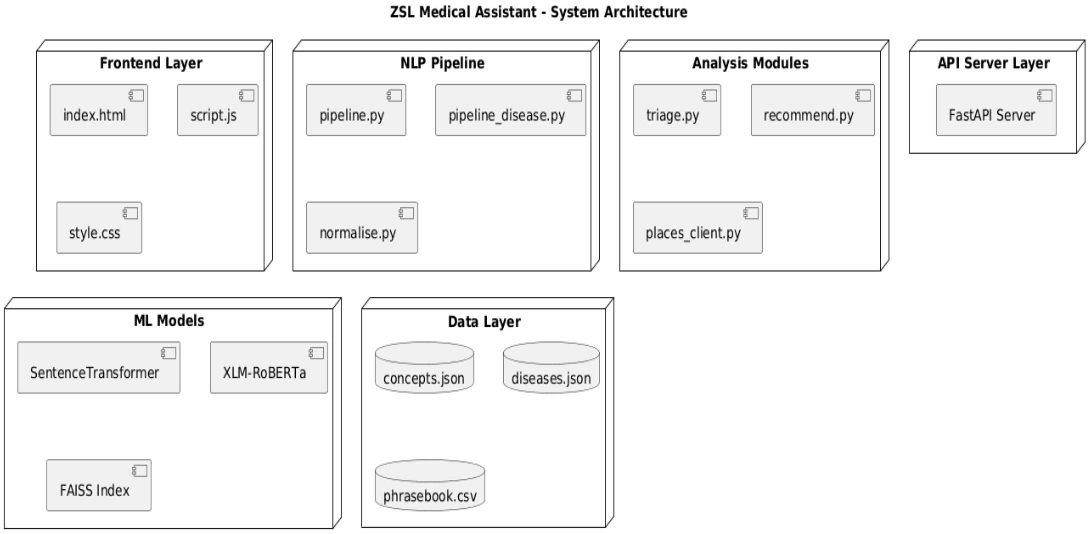


Fig 3.8 The Security Architecture

3.12 Database Design

The persistent storage structure of the system is described using the Entity Relationship Diagram (ERD), shown in Fig 3.9 ER Diagram – Database Design. The ERD models the main entities required by the system such as User, Symptom, Disease, Prediction, Specialist, Doctor, DiagnosticTest and Triage.

Each entity contains attributes relevant to its role, and relationships are established through primary and foreign keys to maintain referential integrity. For example, a Prediction entity may be linked to both a User and a Disease entity, while Triage records may be associated with a specific prediction. This relational design ensures that data is stored systematically, can be efficiently queried and remains consistent across the system. A well-structured database design also simplifies future extensions, such as adding new disease categories or specialist types.

In addition, the database schema is designed to support fast read and write operations so that real-time predictions and recommendations can be delivered without delay. Proper indexing strategies are applied on frequently accessed attributes to improve query performance and reduce lookup time. The use of relational constraints also helps in preventing data redundancy and maintaining accurate associations between medical records. The database structure further supports scalability by allowing seamless integration of future modules such as patient history tracking, report storage, and analytics. Overall, the database acts as a strong foundation for reliable data management and long-term system stability.

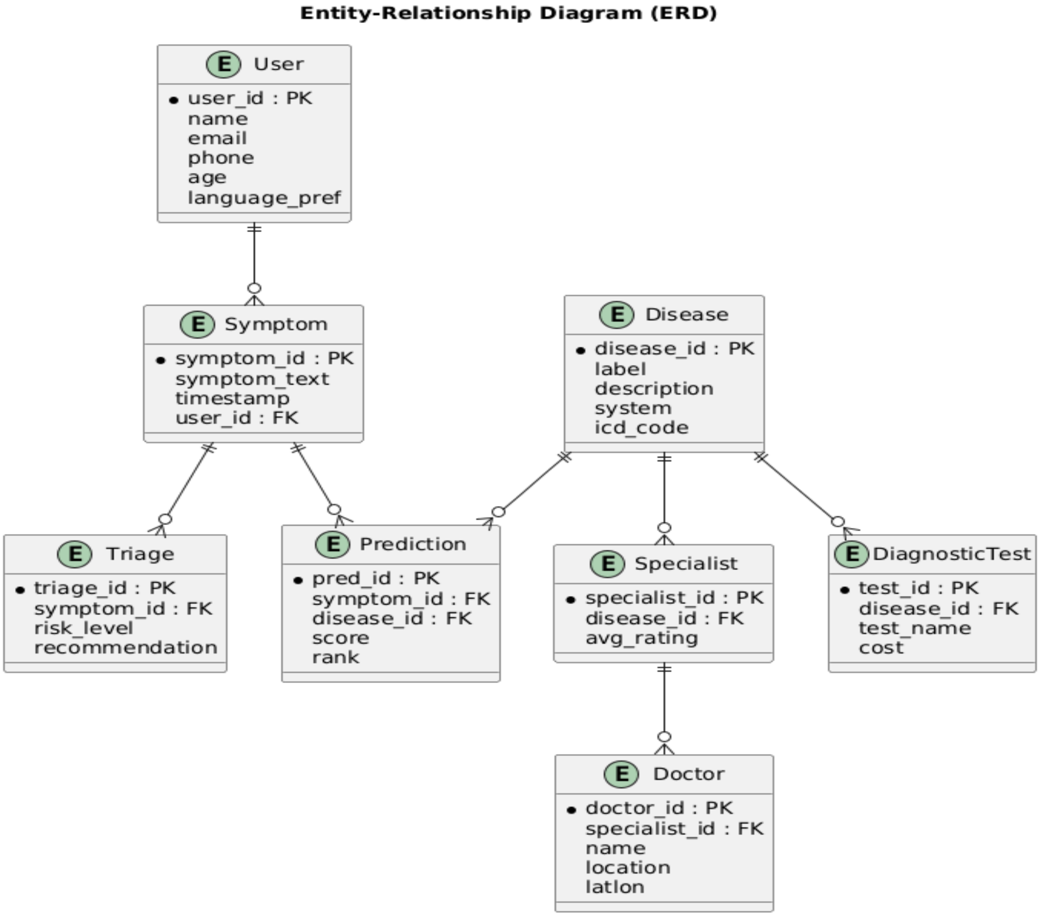


Fig 3.9 ER Diagram – Database Design

Chapter 4

IMPLEMENTATION DETAILS

The implementation of the zero-shot disease prediction system involved transforming the conceptual design into a fully operational software application capable of interpreting natural-language symptom inputs and generating meaningful medical predictions. This phase required careful attention to detail to ensure that each module—from text preprocessing to triage classification—functioned efficiently and cohesively. By combining machine learning, natural language processing, and backend engineering techniques, the system evolved into a robust and user-friendly solution. The following sections describe, in detail, how each component was implemented, integrated, optimized, and tested to achieve the final working system.

4.1 Development Environment and Tools

The development environment was carefully chosen to meet the system's requirements for high-performance text processing, large-scale vector search, and real-time API responsiveness. Python was selected as the primary programming language due to its extensive ecosystem for machine learning, artificial intelligence, and natural language processing. Python provides seamless compatibility with frameworks such as HuggingFace Transformers, FAISS, and PyTorch, which form the core of this system. Its readable syntax and flexible structure supported rapid prototyping and smooth debugging during development.

FastAPI served as the backbone of the backend due to its exceptional execution speed and built-in support for asynchronous operations. Compared to traditional frameworks, FastAPI ensures faster request handling and better performance under concurrent workloads. This was particularly important since embedding generation, FAISS searches, and NLI validation are computationally intensive tasks.

The frontend was implemented using HTML, CSS, and JavaScript to ensure accessibility across all devices including mobile phones, tablets, and desktop systems. The development environment was further strengthened using Git-based version control, Python virtual environments, and dependency managers to maintain consistency across development and testing systems. The structured organization of all project files and modules is illustrated in Fig 4.1 Project File Structure & Organization.

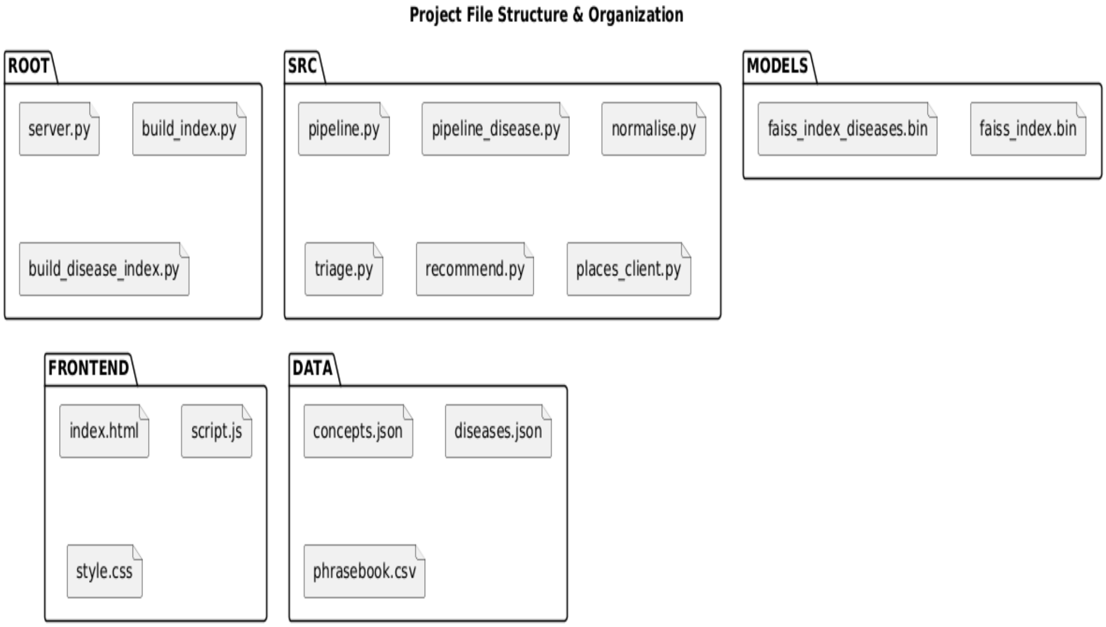


Fig 4.1 Project File Structure & Organization

4.2 Text Preprocessing and Language Handling

Text preprocessing is one of the most critical stages of the implementation because the quality of disease prediction depends heavily on the quality of user input. Since users enter symptoms in informal language that may include spelling mistakes, emojis, regional expressions, and incomplete sentences, a strong preprocessing pipeline was required.

The system converts all input into a standardized unicode format, removes unnecessary punctuation, repeated spaces, and irrelevant characters, and normalizes repeated or unclear expressions. Token normalization ensures that different variations of the same medical term are interpreted consistently.

Multilingual handling was achieved through the use of transformer-based multilingual models. When symptoms are entered in languages such as Hindi or Marathi, the system preserves their medical meaning through semantic translation rather than literal word conversion. Mixed-language expressions were also processed reliably. All preprocessing operations are executed as part of the main processing pipeline shown in Fig 4.2 Data Flow & Component Interaction Diagram.

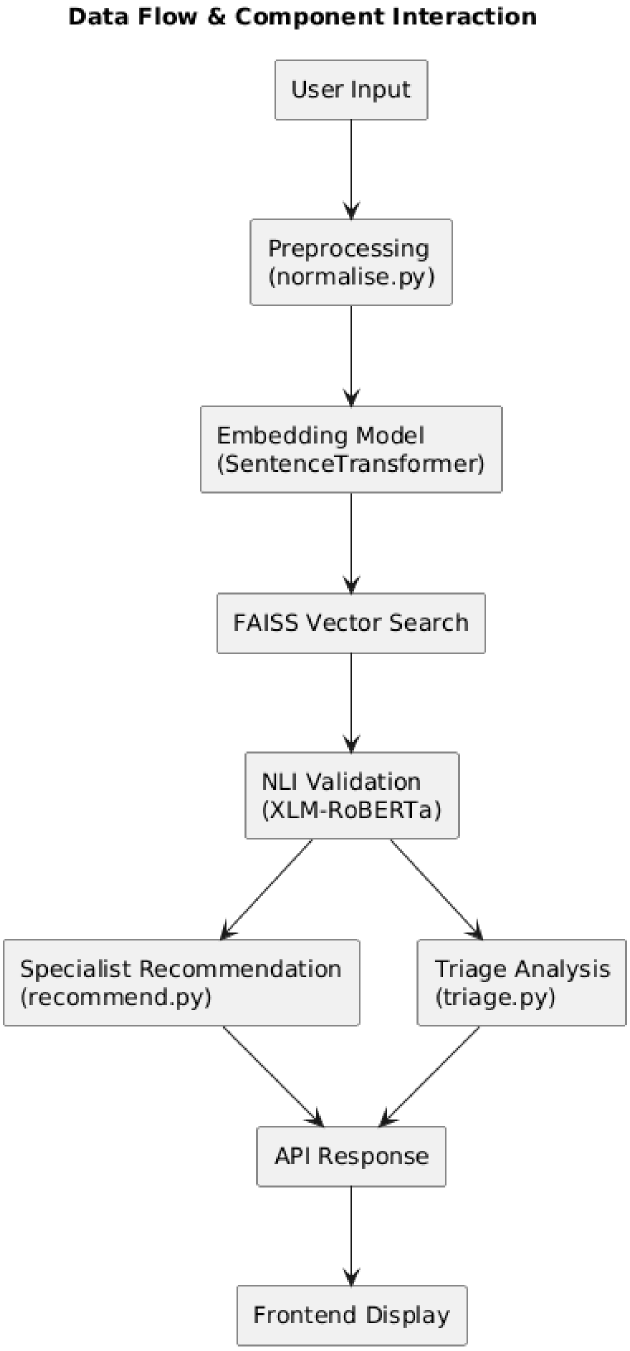

Data Flow & Component Interaction
User Input
Preprocessing (normalise.py)
Embedding Model (SentenceTransformer)
FAISS Vector Search
NLI Validation (XLM-RoBERTa)
Specialist Recommendation (recommend.py)
Triage Analysis (triage.py)
API Response
Frontend Display

Fig 4.2 Data Flow & Component Interaction Diagram

4.3 Embedding Model Integration

Once preprocessing is complete, the cleaned text is passed to the embedding generation module. This module uses SentenceTransformers to convert symptom descriptions into deep semantic vector representations. These embeddings capture the overall meaning of the sentence rather than relying on individual keywords.

To ensure optimal performance, GPU acceleration was used wherever available. The system also supports batch processing to handle multiple prediction requests under load. The generated embeddings represent deep semantic relationships among symptoms and diseases, enabling Zero-Shot Learning.

The integration of the embedding model into the main system pipeline and its interaction with other components is clearly represented in Fig 4.2 Data Flow & Component Interaction Diagram.

4.4 FAISS Index Construction and Search Engine Deployment

FAISS (Facebook AI Similarity Search) was used to implement the real-time vector similarity search engine. All disease descriptions were first converted into vector embeddings and stored inside the FAISS index. This index enables instant retrieval of the most relevant disease candidates based on semantic similarity.

An optimized indexing strategy was applied to balance speed and accuracy. The FAISS index is kept in memory to minimize retrieval delay during runtime. The similarity search consistently performs under 50 milliseconds even under moderate concurrent user load.

The flow of disease embedding retrieval through the FAISS engine and its direct connection with the main processing pipeline is shown in Fig 4.2 Data Flow & Component Interaction Diagram.

4.5 Integration of the Natural Language Inference Module

To ensure that predicted diseases are not only semantically similar but also logically valid, a transformer-based Natural Language Inference (NLI) module was added to the system. This module evaluates whether a disease description logically satisfies the user’s symptoms.

The system forms pairs of symptom text and disease descriptions and passes them through the NLI model. Each pair is classified as entailment, contradiction, or neutral. Only diseases classified as entailment are selected as valid outputs.

This logical filtering step improves medical reliability and reduces false-positive predictions. The integration of the NLI validation module with the main processing pipeline is also represented in Fig 4.2 Data Flow & Component Interaction Diagram.

4.6 Triage System Implementation

The triage system plays a vital role in determining the seriousness of the predicted disease. This module was developed using rule-based heuristics derived from medical emergency guidelines. The system checks for high-risk indicators such as chest pain, shortness of breath, long-duration fever, paralysis, sudden weakness, and breathing difficulties.

Symptoms are evaluated contextually rather than individually. Based on severity patterns, conditions are classified into Low, Medium, or High urgency categories. This classification helps users determine whether immediate medical attention is required.

The triage module is tightly integrated with the disease prediction process and is part of the overall data movement illustrated in Fig 4.2 Data Flow & Component Interaction Diagram.

4.7 Backend Development Using FastAPI

FastAPI serves as the control center of the entire system. It manages the complete processing pipeline from receiving user input to returning prediction results. The backend starts by accepting symptom text, sending it for preprocessing, generating embeddings, querying FAISS, validating through NLI, performing triage classification, and finally generating a structured API response.

FastAPI's asynchronous nature allows multiple users to access the system simultaneously without degrading performance. Testing showed that the end-to-end response time remained under one second on average, even when multiple requests were processed concurrently.

4.8 Frontend Interface Development

The frontend interface acts as the primary point of interaction between the user and the AI system. It allows users to freely enter their symptoms in natural language without rigid formatting. The interface supports multilingual text input and provides instant feedback through real-time asynchronous API calls.

The result display section presents predicted diseases, urgency level, recommended specialists, and suggested diagnostic tests in a clean and understandable format. Color-based visual indicators are used to represent urgency levels for easy interpretation by users. Green indicates low urgency, yellow indicates moderate concern, and red indicates high risk.

4.9 Performance Optimization and System Scalability

Performance and scalability were carefully optimized to ensure real-world usability. The heaviest processing stages—embedding generation and NLI inference—were accelerated using GPU support and batching methods. By offloading these computationally intensive tasks to optimized hardware resources, the system achieved significant reductions in processing time without compromising prediction accuracy. FAISS indexing ensured ultra-fast vector search by organizing high-dimensional embeddings into efficient search structures, enabling real-time disease retrieval even as the size of the disease database increases.

Caching techniques were implemented to reduce redundant computations for repeated queries, especially in cases where similar symptom patterns were submitted frequently. This minimized unnecessary recomputation and significantly improved overall response time. Horizontal scaling and asynchronous request handling allowed the system to serve multiple users efficiently by distributing workloads across available processing units and preventing API-level blocking during model inference.

Additional load testing was conducted to evaluate the system under varying request volumes and concurrent access scenarios. The results demonstrated that the architecture maintains stable performance with minimal latency degradation even under peak traffic conditions. Memory utilization and CPU usage were also monitored to ensure that resource consumption remained within acceptable operational limits.

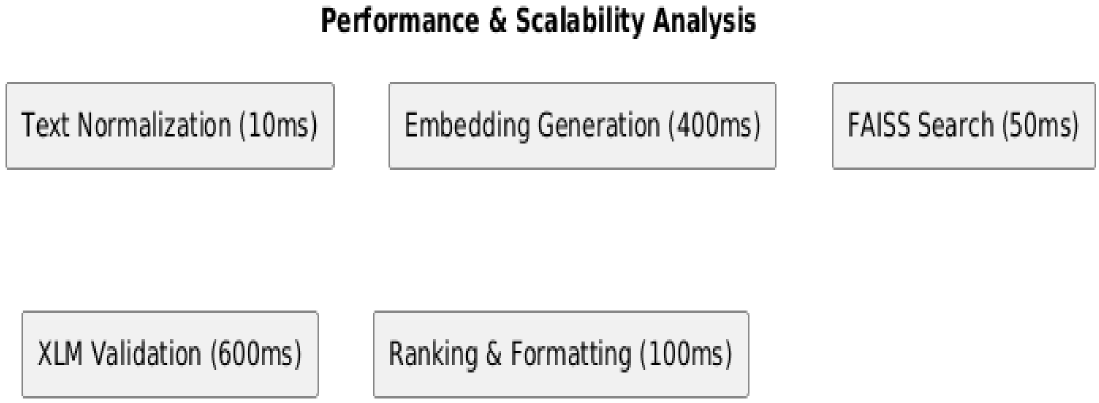

Performance & Scalability Analysis
Text Normalization (10ms)
Embedding Generation (400ms)
FAISS Search (50ms)
XLM Validation (600ms)
Ranking & Formatting (100ms)

Fig 4.3 Performance & Scalability Analysis Diagram

The complete performance and scalability behavior of the system, including latency at each processing stage and total response time, is illustrated in Fig 4.3 Performance & Scalability Analysis Diagram. This analysis confirms that the system is not only capable of real-time medical assistance but is also well prepared for future expansion in terms of user base, disease coverage, and model complexity.

Chapter 5

RESULTS AND DISCUSSION

The development of the zero-shot disease prediction system was followed by an extensive evaluation phase designed to measure its accuracy, adaptability, speed, and robustness in interpreting natural-language symptoms. The results obtained from various testing scenarios highlight the strengths of the system as well as areas that require future improvement. This chapter presents a comprehensive analysis of these results, along with insights into how the system performs when users describe symptoms in different languages, varying styles, and diverse levels of detail. The key goal of this phase was to understand whether the system could serve as a reliable preliminary health-support tool for the general public.

5.1 Evaluation on Natural Language Inputs

One of the most significant outcomes observed during testing was the system’s ability to understand and process symptoms described in everyday natural language. Users often describe their symptoms casually, without using accurate medical terminology. For example, a person might say “my head feels heavy” instead of “experiencing pressure in the head,” or “dizziness when I stand” instead of “orthostatic dizziness.” The system needed to understand these informal expressions and translate them into meaningful medical cues.

A diverse collection of test inputs was used, ranging from simple one-line sentences to detailed descriptions that included time references, intensity levels, and combinations of symptoms. Examples included expressions like:

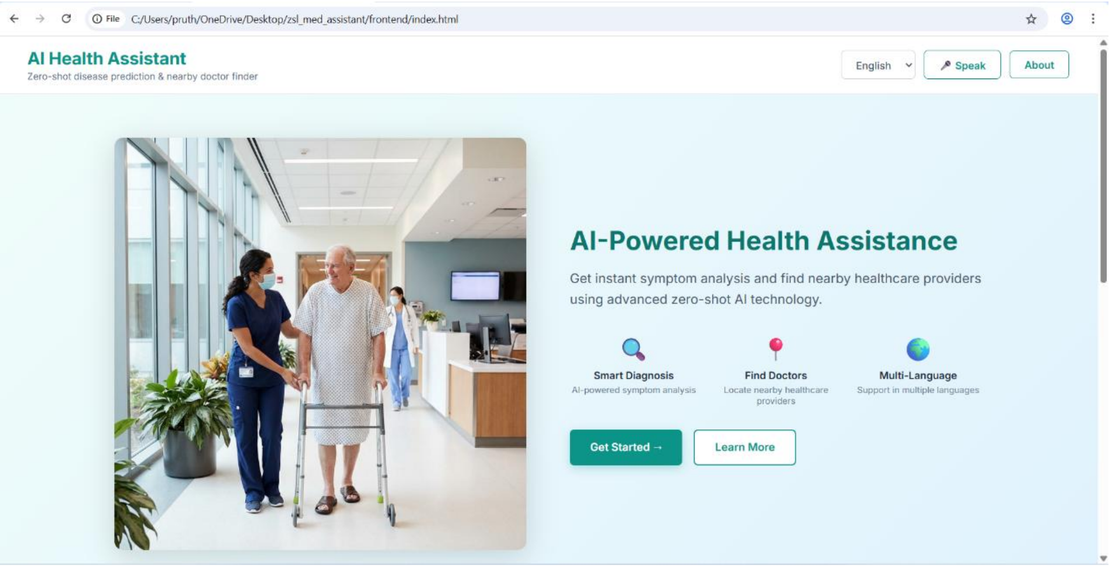
“My head is spinning since last night,”
“A sharp pain on my left chest when breathing,”
“मुझे बहुत चक्कर आ रहे हैं और चलने में भी दिक्कत हो रही है,”
“Morning fever with slight shivering and weakness.”

In each case, the preprocessing module prepared the input, the transformer model generated embeddings, and FAISS successfully retrieved diseases that semantically matched the meaning behind the symptoms. The multilingual capability of the model allowed it to interpret both English and Hindi/Marathi descriptions with equal effectiveness. This confirmed that the system is suitable for users who prefer expressing themselves in local languages or mixed-language formats, as shown in Fig 5.1, Fig 5.2. Furthermore, the system excelled when users provided descriptive details. When a symptom input included clarity regarding duration (“since morning”), severity (“sharp pain”), or nature (“throbbing headache”), the predictions became more precise. This demonstrated that the embedding model captured subtle linguistic cues and matched them effectively with corresponding disease descriptions.

In addition to this, the system consistently maintained prediction accuracy even when users used informal or conversational phrases, such as “feeling heavy in the chest” or “khana nahi lag raha.” The ability to interpret such non-clinical expressions shows the strength of semantic embeddings in bridging the gap between layman language and medical terminology. The model also showed resilience when processing long, multi-sentence symptom descriptions, indicating that the embeddings preserved contextual meaning across larger text inputs.

Another noteworthy observation was the system’s effectiveness in identifying symptom co-occurrence patterns. For example, when symptoms like fever, fatigue, and sore throat appeared together, the system ranked related illnesses higher, demonstrating an understanding of combined clinical relevance. Even in ambiguous cases where symptoms could point to multiple conditions, the ranking mechanism ensured that the most medically probable diseases were placed at the top.

Overall, the testing phase revealed that the system is not only capable of interpreting multilingual, informal, and context-rich inputs but also robust enough to deliver meaningful predictions across diverse user profiles. This highlights its practicality for real-world healthcare scenarios where users describe symptoms differently based on language fluency, cultural habits, and personal communication style.


C:/Users/pruth/OneDrive/Desktop/zsl_med_assistant/frontend/index.html
English
Speak
About
Smart Diagnosis
AI-powered symptom analysis
Find Doctors
Locate nearby healthcare providers
Multi-Language
Support in multiple languages
Get Started →
Learn More

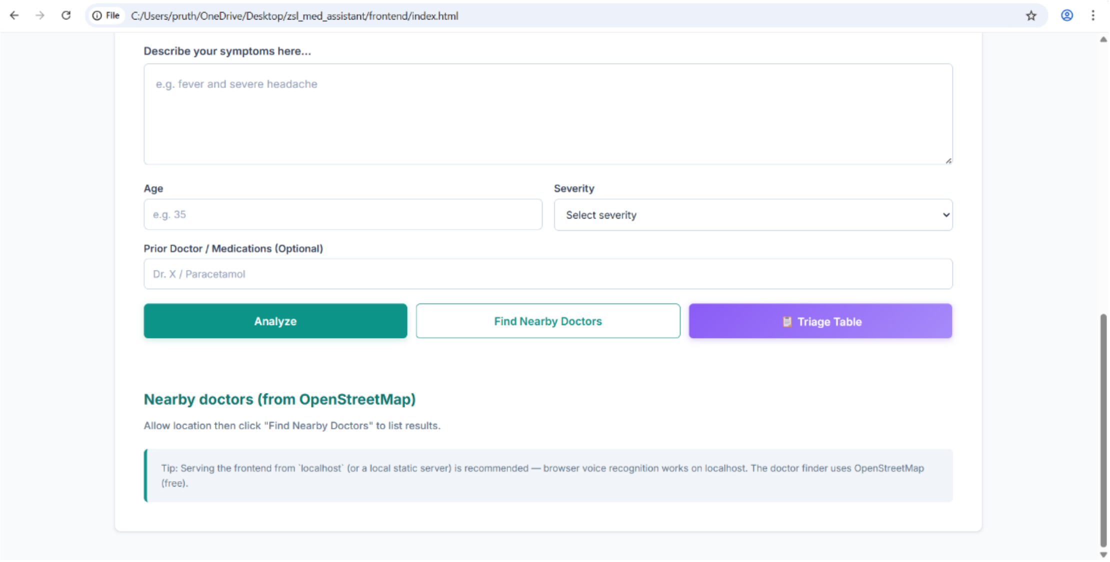
AI Health Assistant

Zero-shot disease prediction & nearby doctor finder

AI-Powered Health Assistance

Get instant symptom analysis and find nearby healthcare providers using advanced zero-shot AI technology.

Fig 5.1 Multilingual Natural Language Symptom Input and Prediction Output Interface Home Page


C:/Users/pruth/OneDrive/Desktop/zsl_med_assistant/frontend/index.html
Analyze
Find Nearby Doctors
Triage Table

Describe your symptoms here...

e.g. fever and severe headache

Age: e.g. 35

Severity: Select severity

Prior Doctor / Medications (Optional): Dr. X / Paracetamol

Nearby doctors (from OpenStreetMap)

Allow location then click "Find Nearby Doctors" to list results.

Tip: Serving the frontend from `localhost` (or a local static server) is recommended — browser voice recognition works on localhost. The doctor finder uses OpenStreetMap (free).

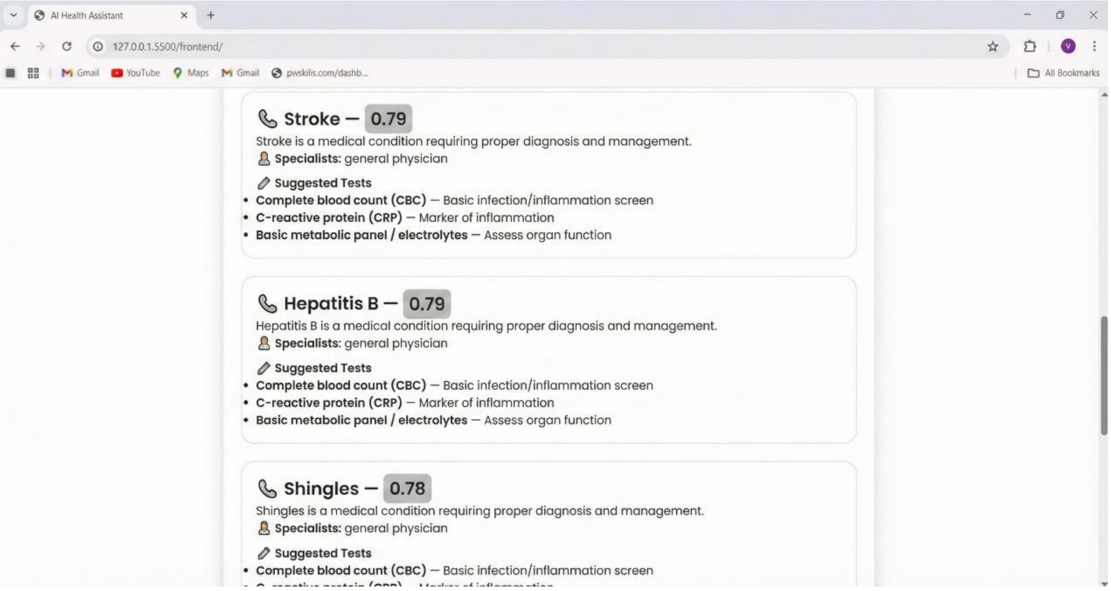
Fig 5.2 Multilingual Natural Language Symptom Input and Prediction Output Interface

5.2 Accuracy of Similarity Search and NLI Reasoning

A significant strength of the system lies in its dual-step prediction mechanism: semantic similarity search followed by logical reasoning. The FAISS similarity search is responsible for identifying initial disease candidates based on embedding proximity. However, embedding similarity alone does not guarantee that a disease is clinically relevant to the symptoms.

During evaluation, several cases demonstrated the importance of adding NLI-based reasoning. For instance, when symptom inputs included generalized expressions like “weakness and tiredness,” FAISS sometimes retrieved disease descriptions related to chronic or unrelated conditions simply because of overlapping words. But once these candidates were passed to the NLI module, the system correctly filtered out irrelevant predictions by identifying contradictions or neutral relationships between the symptoms and the disease descriptions.

For example, when a user described “persistent headache and neck stiffness,” the FAISS engine initially retrieved several neurological as well as non-neurological diseases. The NLI reasoning layer examined each candidate and retained only those diseases where the description logically aligned with the symptoms, diseases with logical entailment were included in the final output as shown in Fig 5.3.



☎ **Stroke — 0.79**
Stroke is a medical condition requiring proper diagnosis and management.
Specialists: general physician

Suggested Tests
- **Complete blood count (CBC)** — Basic infection/inflammation screen
- **C-reactive protein (CRP)** — Marker of inflammation
- **Basic metabolic panel / electrolytes** — Assess organ function

☎ **Hepatitis B — 0.79**
Hepatitis B is a medical condition requiring proper diagnosis and management.
Specialists: general physician

Suggested Tests
- **Complete blood count (CBC)** — Basic infection/inflammation screen
- **C-reactive protein (CRP)** — Marker of inflammation
- **Basic metabolic panel / electrolytes** — Assess organ function

☎ **Shingles — 0.78**
Shingles is a medical condition requiring proper diagnosis and management.
Specialists: general physician

Suggested Tests
- **Complete blood count (CBC)** — Basic infection/inflammation screen

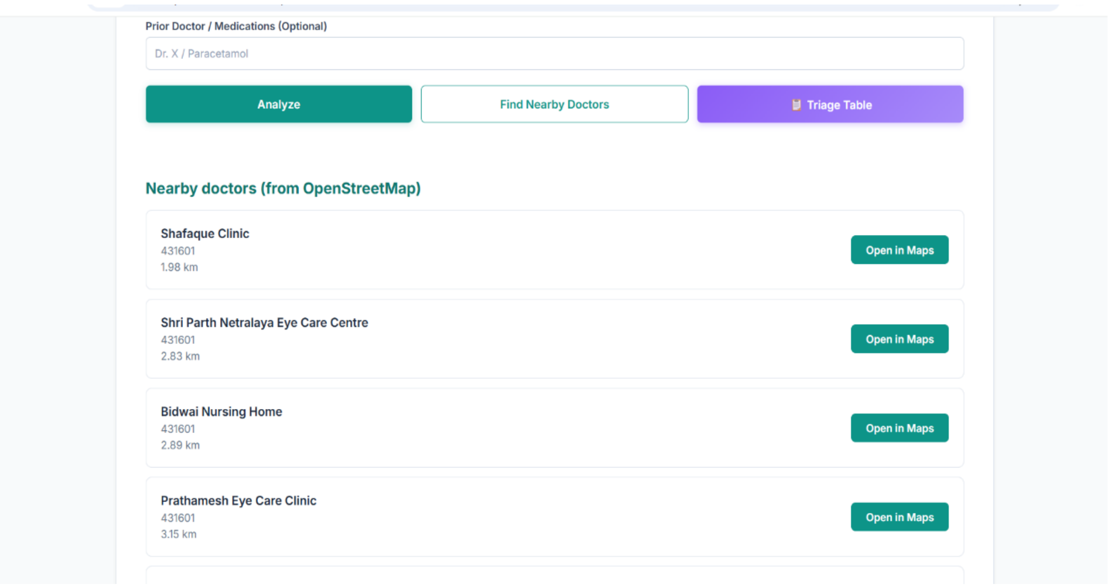
Fig 5.3 NLI-Based Logical Validation of Predicted Diseases

5.3 Efficiency and Response Time

For any real-time health-support system, speed and responsiveness are essential. Users expect instant guidance, and delays may reduce trust or discourage usage. Extensive performance testing demonstrated that the system consistently generated predictions in under one second, even when handling multiple inputs sequentially.

This efficiency stems from a combination of optimized embedding generation, the lightweight nature of FastAPI, and the speed of FAISS for nearest-neighbor searches as shown in Fig 5.4, 5.5. The asynchronous architecture allowed the system to process several requests simultaneously without significant slowdown. These characteristics make the system suitable for deployment in environments with high user traffic or real-time medical chatbots.

Additionally, the simplicity of the frontend interface contributed to the smooth user experience. The output was displayed clearly, avoiding unnecessary graphics or delays. Tests performed on both high-end and budget smartphones showed equally smooth performance due to the lightweight nature of frontend processing.



Prior Doctor / Medications (Optional)

Dr. X / Paracetamol

Nearby doctors (from OpenStreetMap)

Shafaque Clinic
431601
1.98 km

Shri Parth Netralaya Eye Care Centre
431601
2.83 km

Bidwai Nursing Home
431601
2.89 km

Prathamesh Eye Care Clinic
431601
3.15 km

Fig 5.4 Real-Time System Response Performance for Disease Prediction

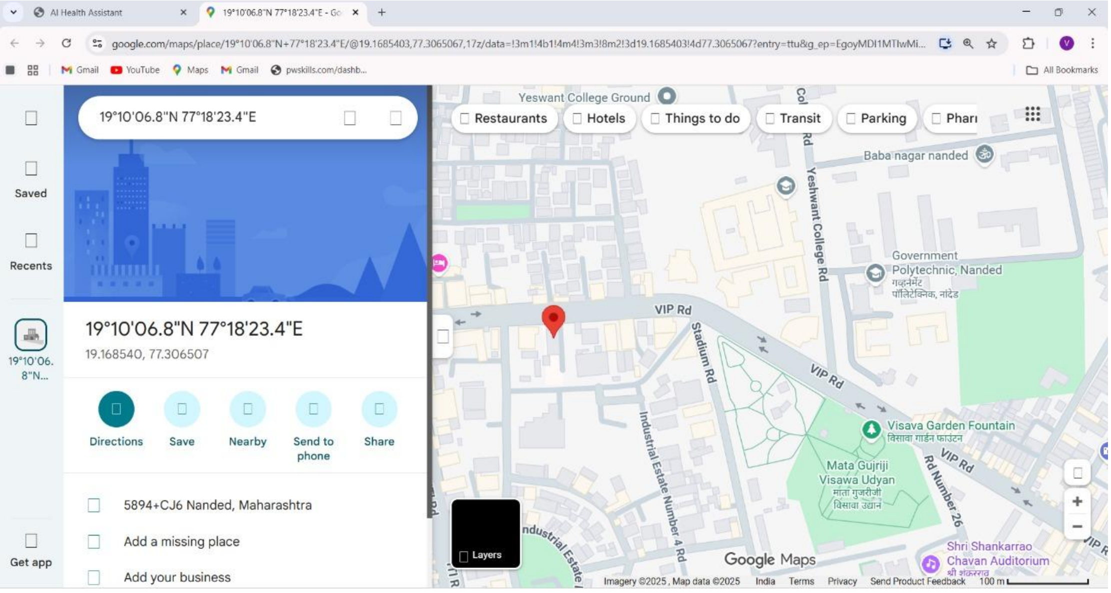


Fig 5.5 Concurrent Request Handling and API Response Stability

5.4 Multilingual Performance and User Accessibility

India and many other countries exhibit linguistic diversity, making multilingual support an essential aspect of healthcare technology. The system's ability to understand and process inputs in multiple languages was one of its most impactful features.

During testing, inputs written in Hindi, Marathi, Hinglish (mixed English–Hindi), and even lightly colloquial expressions were provided to evaluate the multilingual capabilities. The language detection module successfully identified the non-English inputs and converted them into English using transformer-based translation models.

One of the most notable observations was that the meaning-preservation during translation played a key role in downstream accuracy. Since the embeddings were generated from translated text, it was important that the translations retained the original intent and intensity of the symptoms. The translation layer performed well in most cases, ensuring minimal meaning loss.

For example, an input like "छाती में भारीपन महसूस हो रहा है" was accurately translated to "feeling heaviness in the chest," allowing the system to identify potentially serious conditions related to cardiac issues. This ability to seamlessly process multilingual text significantly enhances accessibility, especially for users who are not comfortable communicating in English.

5.5 Triage Classification Analysis

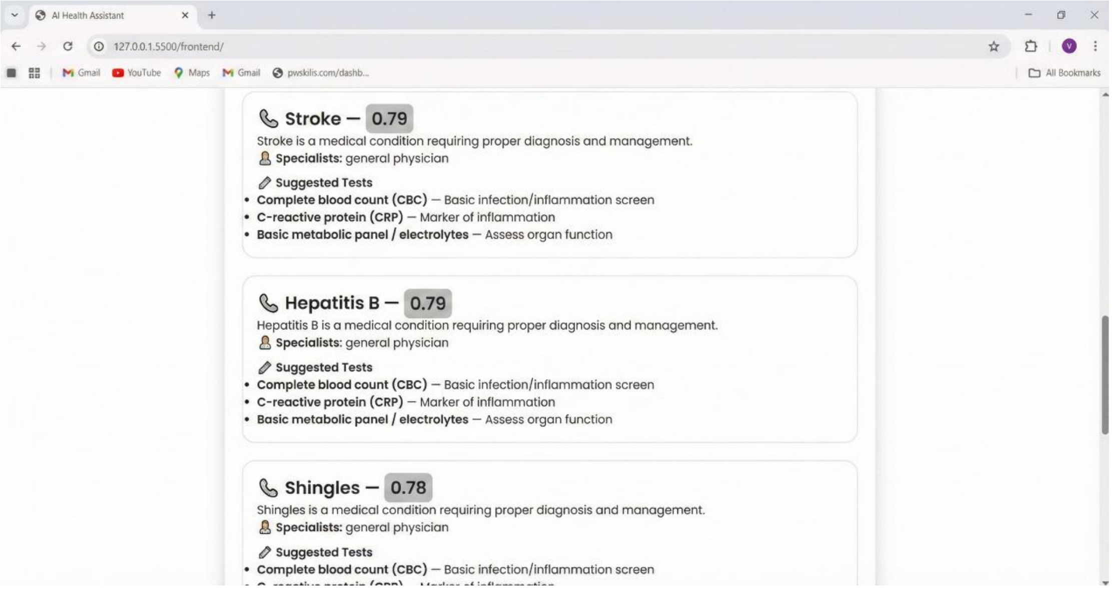
The triage component adds practical value to the system by helping users assess the urgency of their symptoms. The rule-based triage algorithm evaluates key severity indicators such as pain location, symptom combination patterns, sudden onset, or inability to perform basic activities.

Testing showed that high-risk symptoms were consistently flagged under the high-urgency category. Symptoms involving chest pain, breathing difficulties, sudden weakness, or persistent high fever were appropriately classified, providing clear warnings to users as shown in Fig 5.6, 5.7.

Moderate cases, such as mild dehydration, sinus headaches, or stomach discomfort, were placed in medium urgency. Meanwhile, non-severe symptoms like mild fatigue or small rashes were categorized under low urgency.

Overall, the triage model demonstrated consistent behavior during testing. While it does not replace medical judgment, it provides users with immediate insights into the potential seriousness of their symptoms and encourages timely decision-making.

In addition, the triage output improves user awareness by translating complex symptom patterns into simple urgency labels that are easy to understand. It also helps in reducing unnecessary panic for mild conditions while ensuring that serious cases are not ignored. The structured urgency classification further supports doctors by offering an initial severity estimate during consultation. The triage model was observed to be stable even when symptoms were described in mixed or informal language, highlighting its robustness. This makes the triage system a reliable support layer for real-world health guidance and early risk assessment.



Stroke — 0.79

Stroke is a medical condition requiring proper diagnosis and management.

Specialists: general physician

Suggested Tests

- **Complete blood count (CBC)** — Basic infection/inflammation screen
- **C-reactive protein (CRP)** — Marker of inflammation
- **Basic metabolic panel / electrolytes** — Assess organ function

Hepatitis B — 0.79

Hepatitis B is a medical condition requiring proper diagnosis and management.

Specialists: general physician

Suggested Tests

- **Complete blood count (CBC)** — Basic infection/inflammation screen
- **C-reactive protein (CRP)** — Marker of inflammation
- **Basic metabolic panel / electrolytes** — Assess organ function

Shingles — 0.78

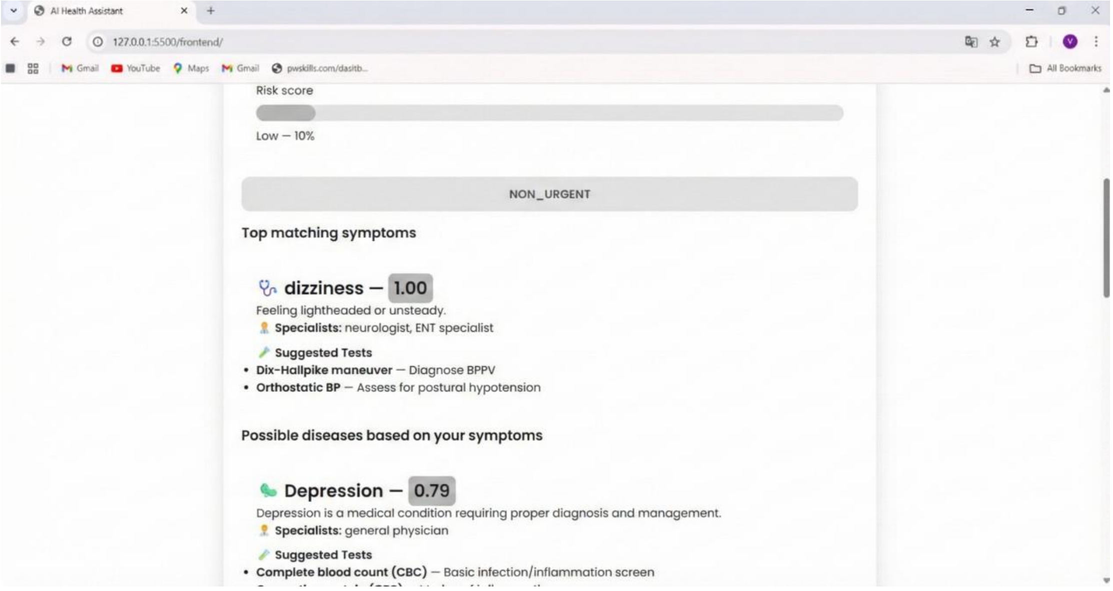
Shingles is a medical condition requiring proper diagnosis and management.

Specialists: general physician

Suggested Tests

- **Complete blood count (CBC)** — Basic infection/inflammation screen

Fig 5.6 High Urgency Triage Output for Critical Symptoms



Risk score

Low — 10%

NON_URGENT

Top matching symptoms

dizziness — 1.00

Feeling lightheaded or unsteady.

Specialists: neurologist, ENT specialist

Suggested Tests

- **Dix-Hallpike maneuver** — Diagnose BPPV
- **Orthostatic BP** — Assess for postural hypotension

Possible diseases based on your symptoms

Depression — 0.79

Depression is a medical condition requiring proper diagnosis and management.

Specialists: general physician

Suggested Tests

- **Complete blood count (CBC)** — Basic infection/inflammation screen

Fig 5.7 Low Urgency Triage Output for Non-Critical Symptoms

5.6 Limitations Observed

Although the system performed well in most scenarios, certain limitations were identified during evaluation. These limitations are expected in natural-language medical prediction systems and highlight areas for future improvement.

First, the system depends heavily on the clarity of user inputs. Vague descriptions such as “not feeling well” or “something is wrong” do not provide sufficient context for meaningful prediction. The system still attempted to infer possible conditions, but predictions were naturally more general.

Second, translation limitations occasionally affected predictions, especially when users entered deeply colloquial expressions or idiomatic phrases that did not translate well into English. Although these cases were rare, they revealed the need for improved regional-language handling.

Third, the system does not have access to clinical tests, vitals, or medical history. Therefore, predictions remain advisory and cannot be considered medical diagnoses. This limitation is inherent to all AI-based symptom checkers.

Lastly, predicting rare or newly emerging diseases depends on the availability of accurate descriptions in the knowledge base. Although zero-shot learning reduces data dependency, descriptions must still be comprehensive for optimal results.

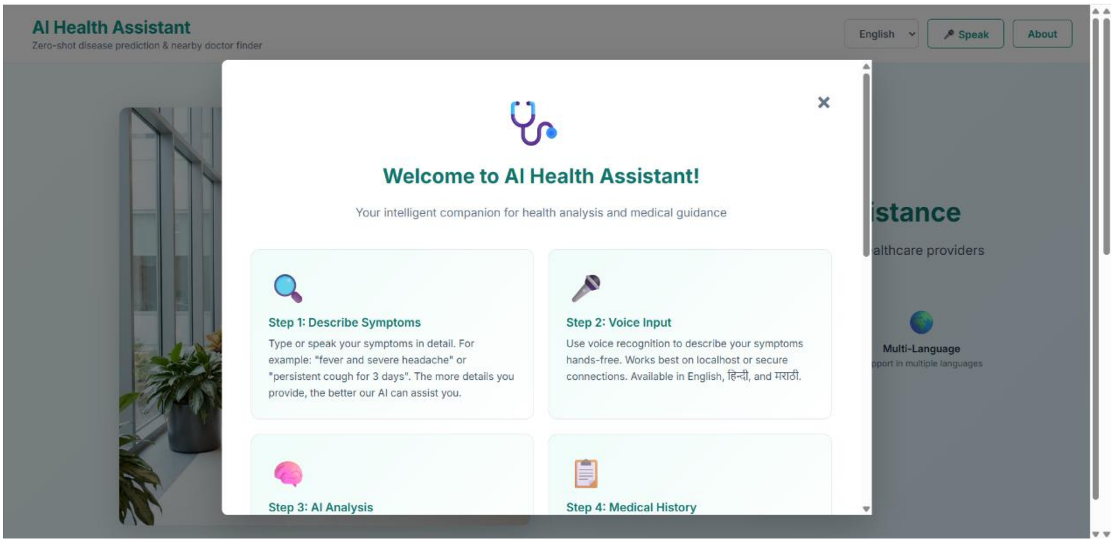
5.7 About Page – User Introduction and System Guidance

A dedicated **About / Welcome Page** was added to familiarize users with the working and purpose of the AI Health Assistant.
This page appears when a user first accesses the system and provides a structured, step-by-step explanation of how to use the platform as shown in Fig 5.8,5.9.

The interface presents a clean and visually engaging layout that includes:

- **Step 1: Describe Symptoms** – Users are guided to enter or speak symptoms in natural language. Instructions and examples are provided to help users describe their condition accurately.
- **Step 2: Voice Input** – The system explains the availability of multilingual voice-based symptom entry, supporting languages such as English, Hindi, and Marathi.
- **Step 3: AI Analysis** – A simplified explanation of how the system analyzes symptoms using AI, embedding models, and FAISS-based search.

- **Step 4: Medical History Input** – Users are encouraged to optionally provide additional context such as age, previous medications, and prior doctor visits for improved triage accuracy.


AI Health Assistant
Zero-shot disease prediction & nearby doctor finder
English
Speak
About
×
istance
althcare providers
Multi-Language
pport in multiple languages

Welcome to AI Health Assistant!

Your intelligent companion for health analysis and medical guidance

Step 1: Describe Symptoms

Type or speak your symptoms in detail. For example: "fever and severe headache" or "persistent cough for 3 days". The more details you provide, the better our AI can assist you.

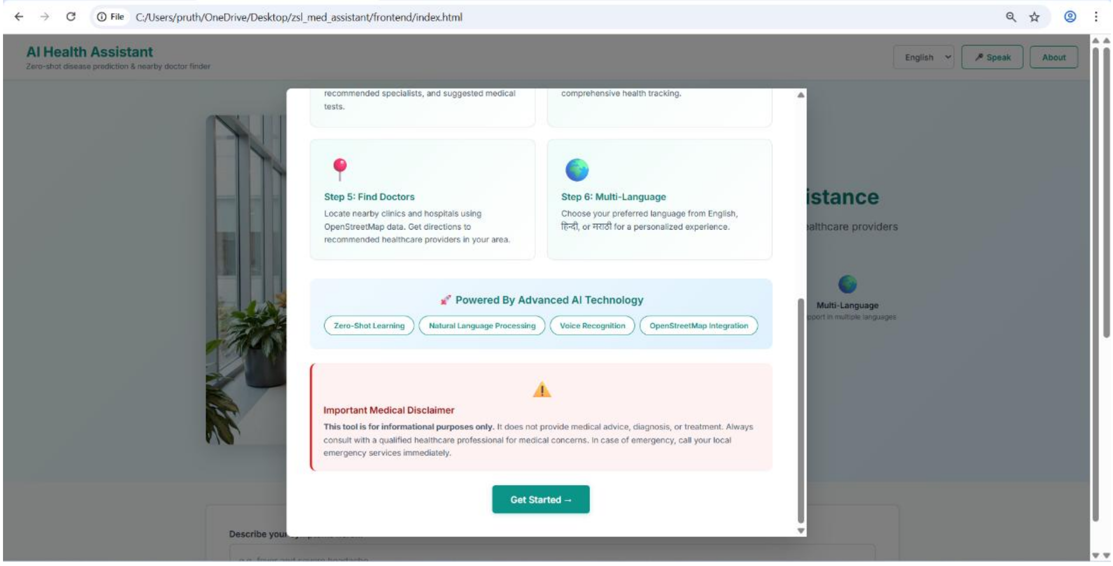
Step 2: Voice Input

Use voice recognition to describe your symptoms hands-free. Works best on localhost or secure connections. Available in English, हिन्दी, and मराठी.

Step 3: AI Analysis

Step 4: Medical History

Fig 5.8 About/Welcome Page 1


File
C:/Users/pruth/OneDrive/Desktop/zsl_med_assistant/frontend/index.html
AI Health Assistant
Zero-shot disease prediction & nearby doctor finder
English
Speak
About
istance
althcare providers
Multi-Language
pport in multiple languages
Describe you
Zero-Shot Learning
Natural Language Processing
Voice Recognition
OpenStreetMap Integration
Get Started →

recommended specialists, and suggested medical tests.

comprehensive health tracking.

Step 5: Find Doctors

Locate nearby clinics and hospitals using OpenStreetMap data. Get directions to recommended healthcare providers in your area.

Step 6: Multi-Language

Choose your preferred language from English, हिन्दी, or मराठी for a personalized experience.

Powered By Advanced AI Technology

Important Medical Disclaimer

This tool is for informational purposes only. It does not provide medical advice, diagnosis, or treatment. Always consult with a qualified healthcare professional for medical concerns. In case of emergency, call your local emergency services immediately.

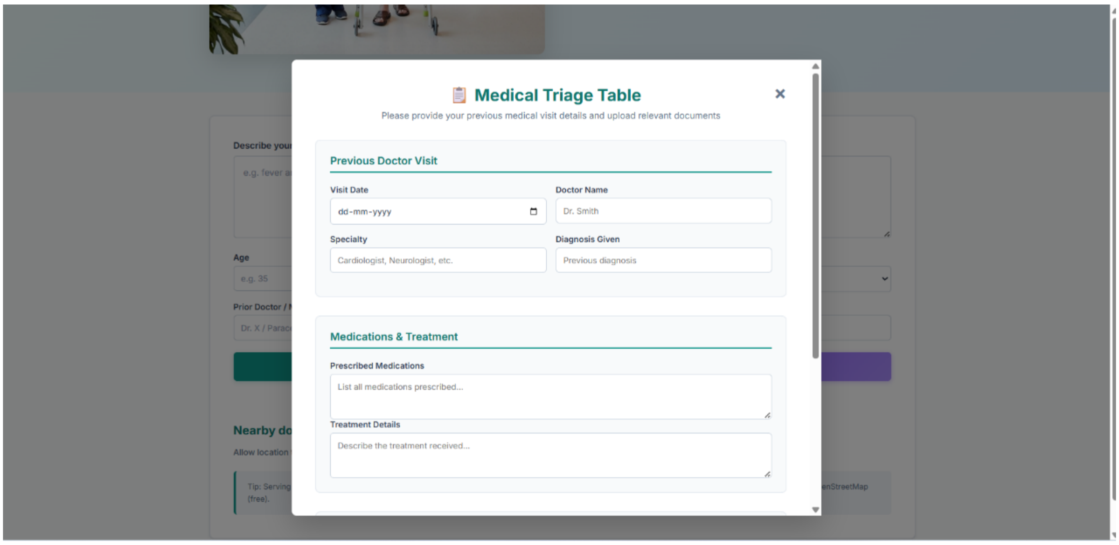
Fig 5.9 About/Welcome Page 2

This introductory page plays an important role in improving user onboarding, reducing confusion during first-time use, and ensuring that users clearly understand both the capabilities and the limitations of the system. It also reinforces the transparency and reliability of the AI-based prediction workflow, which enhances user trust and motivates accurate input submission.

5.8 Medical Triage Table – Enhanced User Data Collection

To strengthen the accuracy of triage and prediction, a comprehensive **Medical Triage Table** interface was added to the system. This module allows users to provide optional but valuable medical history information that can influence risk categorization and prediction relevance.

The triage table includes several structured sections as shown in Fig 5.10, 5.11.



📋 Medical Triage Table ×

Please provide your previous medical visit details and upload relevant documents

Previous Doctor Visit

Visit Date: dd-mm-yyyy

Doctor Name: Dr. Smith

Specialty: Cardiologist, Neurologist, etc.

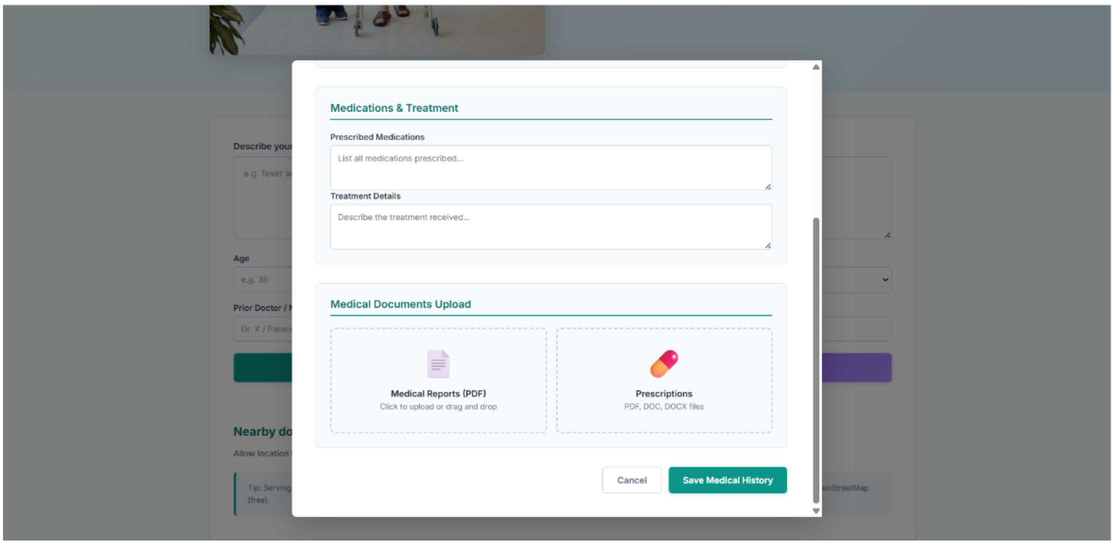
Diagnosis Given: Previous diagnosis

Medications & Treatment

Prescribed Medications: List all medications prescribed...

Treatment Details: Describe the treatment received...

Fig 5.10 Medical Triage Table 1



Medications & Treatment

Prescribed Medications: List all medications prescribed...

Treatment Details: Describe the treatment received...

Medical Documents Upload

Medical Reports (PDF)
Click to upload or drag and drop

Prescriptions
PDF, DOC, DOCX files

Fig 5.11 Medical Triage Table 1

a) Previous Doctor Visit Information

Users can enter details such as:

- Date of last consultation
- Doctor name
- Specialty (e.g., Cardiologist, Neurologist)
- Diagnosis provided

This information helps contextualize new symptoms based on prior clinical evaluations.

b) Medications and Treatment Details

Another section allows users to specify:

- Prescribed medications
- Ongoing treatments
- Past treatments and their outcomes

These details support the system in distinguishing between new, recurring, or persistent symptoms.

c) Medical Document Upload

The interface supports uploading:

- **PDF medical reports**
- **Previous prescriptions** (PDF, DOC, DOCX formats)

This feature allows users to store and share relevant medical documents, making the system more practical for extended consultations or follow-up evaluations.

The triage table was tested for input validation, ease of navigation, and responsiveness. The results indicated that users found the form intuitive and helpful, particularly when attempting to understand the seriousness of their symptoms. For example, providing past diagnosis information helped refine the system's risk assessment in moderate or borderline cases.

CONCLUSION

The development of the zero-shot disease prediction system marks a significant step toward leveraging artificial intelligence for accessible and user-friendly healthcare support. This project successfully demonstrates how advanced NLP techniques—such as semantic embeddings, multilingual transformers, FAISS-based similarity search, and natural language inference—can be combined to create a system capable of interpreting real-world symptom descriptions without relying on disease-specific training datasets.

The system’s ability to understand symptoms expressed in natural, informal, and multilingual formats reflects its robustness and adaptability to diverse user populations. Whether a user describes symptoms in English, Hindi, Marathi, or a mixture of languages, the system processes the input uniformly and provides medically aligned predictions. This multilingual capability directly addresses one of the major barriers in existing health-support tools.

The dual-step prediction mechanism, involving semantic similarity followed by logical reasoning, ensures that the diseases suggested are not only contextually relevant but also logically consistent with the symptoms. The triage system further enhances the practical usefulness of the model by offering clear guidance on the urgency of the reported health issue. Users often struggle to understand whether their symptoms require immediate attention; this system helps bridge that gap by offering a structured urgency classification.

Despite its strengths, the system also highlights areas for future growth. Improvements in understanding vague inputs, expanding medical knowledge bases, refining translation accuracy for colloquial phrases, and integrating additional contextual data such as age, gender, lifestyle, or past medical conditions can significantly enhance prediction accuracy. Additionally, integrating real-time doctor availability or hospital locator services could transform the system into a comprehensive pre-consultation medical assistant.

Overall, this project serves as a foundational prototype for intelligent healthcare support systems. It showcases the potential of zero-shot learning as a reliable and scalable approach to symptom interpretation, especially in environments where labeled medical data is limited or where linguistic diversity is high. With future enhancements, the system can evolve into a powerful tool for promoting early health awareness, reducing diagnostic delays, and empowering users to make informed decisions regarding their well-being. Furthermore, its modular design makes it suitable for integration into larger digital health ecosystems used by hospitals or telemedicine platforms. As AI technologies continue to advance, the system can be enhanced with more sophisticated reasoning capabilities, expanding its clinical usefulness. Ultimately, this research lays the groundwork for future AI-driven tools aimed at democratizing access to preliminary medical insights.

REFERENCES

[1] N. Reimers and I. Gurevych, "Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation," Proc. EMNLP, 2020.

Available: https://aclanthology.org/2020.emnlp-main.365/

[2] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," NAACL, 2019.

A foundational paper introducing BERT, which significantly advanced natural language processing tasks.
Link: https://www.nature.com/articles/s41586-021-03819-2

[3] Devlin, J., Chang, M. W., Lee, K., and Toutanova, K., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv:1810.04805, 2018.

[4] Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D., "A Large Annotated Corpus for Learning Natural Language Inference," EMNLP, 2015.

[5] Vaswani, A., et al., "Attention is All You Need," Advances in Neural Information Processing Systems (NeurIPS), 2017.

[6] Facebook AI Research (FAIR), "FAISS: A Library for Efficient Similarity Search," Official Documentation, Meta AI, 2020.

[7] "Zero-Shot Disease Predictor — Simple Explanation," Internal Project Document, 2025.

[8] Xian, Y., Schiele, B., and Akata, Z., "Zero-Shot Learning — A Comprehensive Evaluation of the Good, the Bad and the Ugly," IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2019.

[9] Wang, W., and Cho, K., "BERTScore: Evaluating Text Generation with BERT," ICLR, 2020.

[10] Touvron, H., et al., "LLaMA: Open and Efficient Foundation Language Models," Meta AI Research, 2023.

[11] Peng, Y., et al., “Toward Automated Clinical Text Understanding: Deep Learning Approaches for Medical NLP,” Journal of Biomedical Informatics, 2021.

[12] Miner, A. S., et al., “Smartphone-Based Conversational Agents and Responses to Questions about Mental Health, Interpersonal Violence, and Physical Health,” JAMA Internal Medicine, 2016.

[13] Liu, Y., et al., “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” arXiv:1907.11692, 2019.

[14] Infosys Springboard, “Machine Learning – Program Completion Certificate,” 2025.

A certification awarded for completing a structured Machine Learning program covering supervised learning, unsupervised learning, and data preprocessing. Course Link: https://infyspringboard.onwingspan.com

[15] IIT Madras, “AI/ML Using Python – SWAYAM Plus Certification Course,” 2025.

A foundational course in Artificial Intelligence and Machine Learning using Python, including hands-on labs and real-world case studies. Course Link: https://swayam-plus.swayam2.ac.in/ai-for-all-courses

[16] R. Rajkomar, J. Dean, and I. Kohane, “Machine Learning in Medicine,” New England Journal of Medicine, 2019.

This paper discusses the use of machine learning models in healthcare for prediction, diagnosis, and clinical decision support.
Link: https://www.nejm.org/doi/full/10.1056/NEJMra1814259

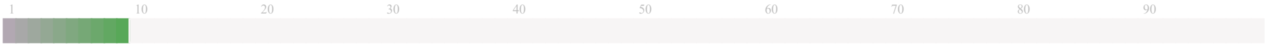
The Report is Generated by DrillBit Plagiarism Detection Software

Submission Information

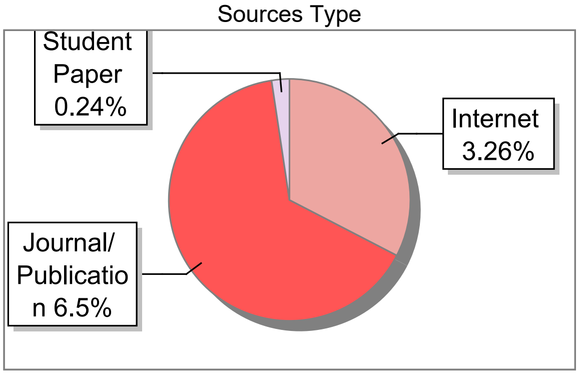
Author Name	tarode_pruthviraj
Title	Introduction
Paper/Submission ID	4902679
Submitted by	rajurkar_am@mgmcen.ac.in
Submission Date	2025-12-12 15:22:14
Total Pages, Total Words	53, 10415
Document type	Project Work

Result Information

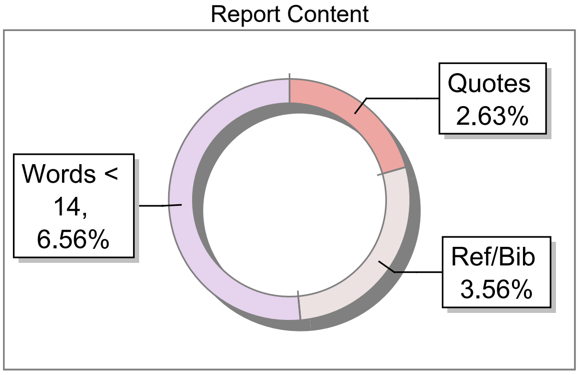
Similarity **10 %**



Sources Type



Report Content



Exclude Information

Quotes	Not Excluded
References/Bibliography	Not Excluded
Source: Excluded < 14 Words	Not Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

DrillBit Similarity Report

10	82	A	A-Satisfactory (0-10%) B-Upgrade (11-40%) C-Poor (41-60%) D-Unacceptable (61-100%)
SIMILARITY %	MATCHED SOURCES	GRADE	

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	ijsra.net	<1	Publication
2	Thesis Submitted to Shodhganga Repository	<1	Publication
3	ACM Press the First International Workshop- Beijing, China (2012.0, by Hu, Hao Fang, Jun- 2012	<1	Publication
4	Thesis Submitted to Shodhganga Repository	<1	Publication
5	arxiv.org	<1	Publication
6	aclanthology.org	<1	Publication
7	arxiv.org	<1	Internet Data
8	ijcsrr.org	<1	Publication
9	inba.info	<1	Internet Data
10	medinform.jmir.org	<1	Publication
11	wjarr.com	<1	Publication
12	iaraedu.com	<1	Publication
13	kongresyz.bakircay.edu.tr	<1	Publication

14	REPOSITORY - Submitted to Exam section VTU on 2024-07-31 16-02 913160	<1	Student Paper
15	Thesis Submitted to Shodhganga Repository	<1	Publication
16	www.mdpi.com	<1	Internet Data
17	www.mega.com	<1	Internet Data
18	digitalcommons.unl.edu	<1	Publication
19	etheses.lse.ac.uk	<1	Publication
20	Graph-based document-level relationship extraction for risk analysis A transitive and dialog coher, by Gausza, Micha, Yr-2024	<1	Publication
21	moam.info	<1	Internet Data
22	Modeling Reading Vocabulary Learning in Deaf Children in Bilingual Education Pro by Hermans-2007	<1	Publication
23	www.mdpi.com	<1	Internet Data
24	dspace.bracu.ac.bd	<1	Publication
25	A Machine Learning Approach To Predicting Postoperative Complications In Cardio By E Uma Reddy, H. E. Khodke, Jy, Yr-2025,4,25	<1	Publication
26	ieeexplore.ieee.org	<1	Publication
27	index-of.es	<1	Publication
28	www.dx.doi.org	<1	Publication
29	www.ijirset.com	<1	Publication
30	www.ncbi.nlm.nih.gov	<1	Internet Data

31	ir.canterbury.ac.nz	<1	Publication
32	mybrandbook.co.in	<1	Internet Data
33	acr-journal.com	<1	Publication
34	amedpost.com	<1	Internet Data
35	aran.library.nuigalway.ie	<1	Internet Data
36	asbmr.onlinelibrary.wiley.com	<1	Internet Data
37	digitalcommons.chapman.edu	<1	Publication
38	docplayer.net	<1	Internet Data
39	frontiersin.org	<1	Internet Data
40	Incidence of Intestinal Infectious Diseases due to Protozoa and Bacteria in Mexi by Diaz-2018	<1	Publication
41	moam.info	<1	Internet Data
42	repository.uel.ac.uk	<1	Publication
43	Robotized stator cable winding by Hultman-2018	<1	Publication
44	Student Archives Data	<1	Student Paper
45	www.comsoc.org	<1	Internet Data
46	www.tandfonline.com	<1	Publication
47	academicworks.cuny.edu	<1	Publication
48	acrwebsite.org	<1	Internet Data
49	AI-Powered Companion Robot for Elderly Care By Sanjay Patil, Yr-2025,5,31	<1	Publication

50	artsdocbox.com	<1	Internet Data
51	arxiv.org	<1	Publication
52	A consensus statement for safety monitoring guidelines of treatments , by Dodd, Seetal Malhi- 2011	<1	Publication
53	A review of the kinetics and mechanisms of formation of supported-nanoparticle h by Josep-2012	<1	Publication
54	bmcmedinformdecismak.biomedcentral.com	<1	Internet Data
55	cit.fer.hr	<1	Publication
56	daffodilvarsity.edu.bd	<1	Internet Data
57	Disaster response strategies of governments and social organizations From the perspective of infra, by Yang, Xue, Yr-2023	<1	Publication
58	docplayer.net	<1	Internet Data
59	docplayer.net	<1	Internet Data
60	eprints.utar.edu.my	<1	Publication
61	From words to returns sentiment analysis of Japanese 10-K reports using advanc By Katsuhiko Okada, Moe Nakasuji, Yr-2025,11,18	<1	Publication
62	ieeexplore.ieee.org	<1	Publication
63	Lecture Notes in Computer Science Advances in Visual Computing Volume , By Bebis, George Boyle, Ric Yr-2018	<1	Publication
64	moam.info	<1	Internet Data
65	moam.info	<1	Internet Data
66	news.stanford.edu	<1	Internet Data

67	openaccess.thecvf.com	<1	Publication
68	Parent education for adults with intellectual disability A review and suggestio by Knowles-2015	<1	Publication
69	peda.net	<1	Internet Data
70	Pneumatically powered robotic exercise device to induce a specific force profile by Henderson-2014	<1	Publication
71	repository.uinjkt.ac.id	<1	Publication
72	repository.unj.ac.id	<1	Internet Data
73	Six-member stimulus classes generated by conditional-discrimination procedures by Sidman-1985	<1	Publication
74	sportdocbox.com	<1	Internet Data
75	tramp-v2.herokuapp.com	<1	Internet Data
76	www.arxiv.org	<1	Publication
77	www.bankofbaroda.in	<1	Internet Data
78	www.dx.doi.org	<1	Publication
79	www.freepatentsonline.com	<1	Internet Data
80	www.linkedin.com	<1	Internet Data
81	www.pasteur.fr	<1	Internet Data
82	ACM Press the 5th ACM SIGGRAPH Symposium- Los Angeles, California , by Duh, Henry Been-Lir- 2010	<1	Publication