Medi Help - AI Based Symptom Checker And Ayurvedic Home Remedy Recommendation

A PROJECT REPORT

Submitted by

Gopi Chand [RA2211026010065]
V Sanjeev Reddy [RA2211026010116]

Under the Guidance of

Dr. M.S.Abirami
Associate Professor,
Department of Computational Intelligence

in partial fulfilment of the requirements for the degree
of
BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING
with specialization in
ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

DEPARTMENT OF COMPUTATIONAL INTELLIGENCE COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR- 603 203

MAY 2025

Department of Computational Intelligence
SRM Institute of Science & Technology
Own Work* Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

Degree/ Course : **B.Tech/Computer Science Engineering in AI and ML**

Student Name : **Gopi Chand, V Sanjeev Reddy**

Registration Number : **RA2211026010065, RA2211026010116**

Title of Work : Medi Help - AI Based Symptom Checker And Ayurvedic Home Remedy Recommendation

We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

We confirm that all the work contained in this assessment is my / our own except where indicated, and that We have met the following conditions:

- Clearly referenced / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:	
I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is our own work, except where indicated by referring, and that I have followed the good academic practices noted above.	
RA2211026010065	RA2211026010116

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that 21CSP302L – Project report titled "**Medi Help - AI Based Symptom Checker And Ayurvedic Home Remedy Recommendation**" is the Bonafide work of " **V Sanjeev Reddy[RA2211026010116], N. Gopi Chand [RA2211026010065]** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. M .S. Abirami
SUPERVISOR
Associate Professor
Department of
Computational Intelligence

SIGNATURE

Dr. R. Annie Uthra
PROFESSOR &HEAD
Department of
Computational Intelligence

EXAMINER 1

EXAMINER 2

ACKNOWLEDGEMENTS

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to **Dr. Leenus Jesu Martin M,** Dean-CET, SRM Institute of Science and Technology, for his invaluable support.

We wish to thank **Dr. Revathi Venkataraman**, Professor and Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We encompass our sincere thanks to, **Dr. M. Pushpalatha**, Professor and Associate Chairperson - CS, School of Computing and **Dr. Lakshmi,** Professor and Associate Chairperson -AI, School of Computing, SRM Institute of Science and Technology, for their invaluable support

We are incredibly grateful to our Head of the Department, Dr. R. Annie Uthra, Professor & Head department of Computational Intelligence, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our Project Coordinators, Panel Head, and Panel Members Department of Computational Intelligence, SRM Institute of Science and Technology, for their inputs during the project reviews and support.

We register our immeasurable thanks to our Faculty Advisor, Dr. K. Babu, Department of Computational Intelligence, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to our guide, Dr. M.S. Abirami, Department of Computational Intelligence, SRM Institute of Science and Technology, for providing us with an opportunity to pursue our project under her mentorship. She provided us with the freedom and support to explore the research topics of our interest. Her passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank all the staff members of Computational Intelligence, School of Computing, S.R.M Institute of Science and Technology, for their help during our project. Finally, we would like to thank our parents, family members, and friends for their unconditional love, constant support and encourageme

ABSTRACT

The increasing demand for accessible and personalized healthcare solutions has prompted the exploration of intelligent systems that combine modern AI techniques with culturally rooted medical practices. This research introduces MediHelp, a hybrid AI-powered system designed to assist users in identifying potential diseases based on their symptoms and personal health profiles, while recommending contextually appropriate home remedies drawn from traditional healing knowledge. The system aims to empower users—particularly in underserved or remote communities—by providing initial diagnostic support and natural treatment options without the immediate need for professional medical intervention.MediHelp accepts user inputs including age, symptoms, dietary preferences, allergies, and pre-existing conditions. It utilizes a machine learning model to predict the most likely disease and matches the result with a curated dataset of home remedies. Each remedy is filtered according to the user's profile to ensure safety and relevance. The system supports structured symptom input via an intuitive interface and includes detailed remedy descriptions covering ingredients, preparation methods, and potential side effects.The architecture is built around a layered client-server model, ensuring modularity, scalability, and ease of integration with future extensions such as user feedback loops or mobile applications. Evaluation results indicate strong predictive performance and remedy relevance, with high usability scores among test users.This work contributes to the growing field of intelligent healthcare systems by demonstrating the feasibility of combining AI diagnostics with ethnomedicinal knowledge. It also highlights the potential of culturally informed, user-centered design in improving digital health accessibility. Future directions include expanding the disease and remedy database, integrating multilingual support, and incorporating explainable AI techniques to increase user trust and transparency. Overall, MediHelp serves as a practical and research-oriented step toward personalized, preventive, and inclusive digital health solutions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

AI	Artificial Intelligence
ML	Machine Learning
NLP	Natural Language Processing
LLM	Large Language Model
EHR	Electronic Health Record
SVM	Support Vector Machine
RF	Random Forest
KNN	K-Nearest Neighbor
RAG	Retrieval-Augmented Generation

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

Within the realm of digital healthcare in the present day, there are a multitude of technologies available for disease prediction determined by symptoms and user history. On the other hand, the majority of these systems ultimately direct customers toward pharmacological solutions. Furthermore, an excessive reliance on medications can result in unfavorable side effects and ignores other medical systems, despite the fact that they are useful in certain circumstances. On the other hand, natural home remedies, many of which have their origins in Ayurveda, provide therapy choices that are more gentle and sustainable for a wide variety of common disorders. This project, which is a web application powered by artificial intelligence and is called MediHelp-AI-Based Symptom Checker and Ayurvedic Home Remedy Recommender, not only makes a prediction about the disease that is most likely to be present based on a user's age, symptoms, allergies, and pre-existing conditions, but it also makes recommendations for individualized home remedies that are suitable for the user's age group and dietary preferences. It offers a healthcare solution that is both balanced and user-friendly by combining modern artificial intelligence with ancient knowledge.

1.2 PROBLEM STATEMENT

Despite the fact that digital symptom checkers are becoming more accessible, they frequently result in a recommendation system that is either biased or inadequate. This system promotes pharmaceutical items and frequently ignores the possibility of treatments that may be performed at home. In addition, the home remedy applications that are currently available do not offer specialized filtering services that take into account dietary requirements, allergies, or age. This gap underlines the need for a system that not only forecasts the likelihood of prospective ailments but also provides recommendations for home remedies that are safe, natural, and individualized by utilizing a database that has been ethically maintained.

1.3 MOTIVATION

The increasing reliance on pharmaceutical treatments for even relatively minor problems generates concerns about the long-term impact on health, the burden on the economy, and the potential adverse repercussions that may develop as a result of this trend. These concerns are prompted by the increasingly widespread use of pharmacological treatments. In order to accomplish the goal of establishing a platform that encourages the utilization of natural medicines as primary forms of therapy for illness, this project was driven by the objective of construction. As a source of inspiration for the project, which was influenced by Ayurvedic principles, a conviction in healing that is based on nature served as a source of inspiration. MedHelp provides an emphasis on therapy that is centered on the patient and supports decisions that are health-conscious by providing tailored ideas for home remedies. In addition, MediHelp encourages patients to focus on their own health. On the other hand, commercial solutions, which often promote branded medications, are not involved in this situation.

1.4 SUSTAINABLE DELEVOPMENT GOALS

Specifically aligned with Sustainable Development Goal 3: Good Health and Well-being, the MediHelp – AI Based Symptom Checker and Ayurvedic Home Remedy Recommender project makes a significant contribution to the Sustainable Development Goals (SDGs) of the United Nations. For people of all ages, this objective places an emphasis on guaranteeing healthy lives and encouraging well-being within the population. The availability of healthcare that is both inexpensive and devoid of adverse effects continues to be a considerable obstacle in many regions of the world, particularly in underdeveloped countries for example. By promoting the utilization of traditional home remedies that are based on Ayurveda and other natural healing traditions, this project provides a method that is both sustainable and inclusive for the treatment of common disorders. These cures frequently make use of chemicals that are easily accessible in the home, which provides them with the advantage of being both affordable and accessible to a large population.

Additionally, the project contributes to Sustainable Development Goal 12 ("Responsible Consumption and Production") likewise. Overuse of pharmaceutical pharmaceuticals and incorrect disposal of these drugs have resulted in degradation of the environment, the development of antibiotic resistance, and increased expenditures associated with medical care. A more responsible model of health consumption is promoted by MediHelp through the provision of an alternative that is based on substances that are derived from natural sources and are biodegradable. Individuals are given the ability to take preventative measures for their health by utilizing remedies derived from nature that are tailored to their food choices, allergy sensitivities, and age. In the end, the initiative acts as a bridge between the growth of technology and sustainable health practices, fostering well-being in a manner that is both environmentally friendly and socially responsible.

CHAPTER 2

LITERATURE SURVEY

2.1 OVERVIEW OF THE RESEARCH AREA

There has been a significant development in the intersection of artificial intelligence and healthcare in recent years, particularly in areas such as the prediction of diseases, the research of symptoms, and digital diagnostics. This expansion has been particularly noticeable in the field of digital diagnostics. Applications like WebMD, Ada Health, and Babylon were among the first to demonstrate the use of artificial intelligence for the purpose of performing symptom checks. The procedure of medical self-assessment is made easier for users by the utilization of these tools. These systems make use of machine learning algorithms, huge medical knowledge libraries, and replies from users in order to provide diagnostic recommendations. The goal of these systems is to provide diagnostic assistance. The vast majority of these platforms, on the other hand, are only focused with diagnosis and treatment options, particularly those that are geared toward pharmaceutical treatments to be more specific. There is a rising interest in combining these alternative approaches into contemporary health applications as a result of the expansion of holistic health movements and the expanding knowledge of traditional medical systems such as Ayurveda. It is also a result of the fact that these systems are becoming more well known.

2.2 EXISTING MODELS AND FRAMEWORKS

Advances in artificial intelligence (AI) and machine learning (ML) have significantly accelerated the development of intelligent healthcare systems for disease prediction and personalized treatment recommendations. Early works, such as those reviewed by Ghaniaviyanto Ramadhan et al. [11], highlighted the importance of proper data preprocessing — including handling missing values, outliers, and class imbalance — to enhance prediction accuracy in chronic disease models.

Nayak et al. [12] proposed an intelligent disease prediction and drug recommendation system combining multiple machine learning approaches with sentiment analysis techniques, showing the value of fusing patient feedback into diagnosis and treatment pipelines. Kim and Joe [13] introduced Symptom2Vec, a deep learning-based method that models symptom relationships and improves automatic medical diagnosis, achieving notable improvements in classification metrics.

Wearable technologies have opened new possibilities for real-time disease prediction. Himi et al. [14] developed MedAi, a smartwatch-based system that predicts multiple diseases using bodily statistics collected from sensors, employing various machine learning algorithms. Similarly, Li and Jha [10] proposed DOCTOR, a continual learning framework capable of detecting multiple diseases dynamically using wearable medical sensors, enabling the system to adapt as new patient data becomes available.

In parallel, large language models (LLMs) have begun to play an influential role in healthcare applications. Wu et al. [3] introduced a hybrid framework combining traditional natural language processing (NLP) with LLMs to improve rare disease phenotyping, demonstrating that LLMs can bridge gaps in knowledge extraction from unstructured clinical text. Cui et al. [4] advanced this further by employing multiple LLM agents (predictor and critic) to enhance few-shot disease prediction capabilities using electronic health records (EHR).

Yu et al. [5] proposed Health-LLM, a personalized disease prediction system that augments LLM reasoning with structured retrieval mechanisms, resulting in improved prediction accuracy through retrieval-augmented generation. Zhao et al. [6] provided a broader perspective by surveying how LLMs, such as GPT variants, are increasingly being integrated into recommender systems, enhancing personalization and reasoning in recommendation scenarios.

Beyond disease prediction, several researchers have focused on drug and remedy recommendation systems. Zhang et al. [15] developed a knowledge-enhanced multi-task learning framework that leverages both biomedical knowledge graphs and molecular attributes to improve medicine recommendation under sparse data conditions. Phan et al. [9] introduced FastRx, a system that combines Fastformer attention mechanisms with memory-augmented graph neural networks for personalized medication recommendation, achieving competitive performance.

From a molecular interaction perspective, Zhao et al. [7] proposed DEPOT, a motif-aware recommendation system that decomposes drug molecules into functional motifs to better understand drug-disease relationships. Zou et al. [8] designed DAI-Net, a dual adaptive interaction network that models both coarse-grained and fine-grained interactions between patient symptoms and medication substructures, resulting in improved coordinated medication recommendations.

Moreover, the integration of emotional and behavioral feedback in health recommender systems has gained attention. Nayak et al. [12] utilized NLP-based sentiment analysis to refine drug recommendations based on user reviews, showcasing the potential of user-centric system adaptation.

In summary, the convergence of structured medical data processing, wearable sensing technologies, graph-based learning, and LLM-driven reasoning is redefining disease prediction and remedy recommendation frameworks. Despite substantial progress, as highlighted by these studies, there remains a significant opportunity to expand these techniques into culturally sensitive domains such as Ayurvedic remedy recommendation, which remains largely underexplored.

2.3 LIMITATIONS IDENTIFIED FROM LITERATURE SURVEY

Despite major advancements in AI-driven healthcare systems, significant gaps persist that limit the applicability of existing models for Ayurvedic and traditional healthcare practices.

Most disease prediction systems [11], [12], [13] focus on structured biomedical datasets and overlook holistic symptom profiles essential to Ayurvedic diagnosis. Similarly, real-time monitoring frameworks utilizing wearable sensors [10], [14] concentrate on physiological data without integrating culturally specific indicators such as body constitution (Prakriti) or seasonal imbalances emphasized in Ayurveda.

While recent research into LLM-based medical reasoning [3], [4], [5], [6] shows promise, current models are primarily trained on contemporary medical literature and fail to reason over ancient Ayurvedic texts or indigenous knowledge bases. This restricts their usability for culturally relevant healthcare support.

Furthermore, drug and remedy recommendation systems [7], [8], [9], [15] are confined to allopathic medicines and lack personalized Ayurvedic remedy suggestion mechanisms. Emotional feedback integration in recommender systems [12] also remains centered on western healthcare treatments, neglecting traditional remedy experiences.

Thus, there exists a critical research opportunity to build a system that:

- Predicts diseases using holistic, symptom-based models aligned with Ayurvedic principles.
- Recommends personalized Ayurvedic home remedies filtered by user-specific factors like age, allergies, and dietary preferences.
- Leverages LLMs capable of understanding and reasoning over traditional healthcare knowledge.
- Integrates user feedback on traditional remedies to refine recommendation accuracy.

Filling this gap would significantly contribute to the development of culturally sensitive, AI-assisted healthcare solutions, bridging modern technology with ancient medical wisdom.

2.4 RESEARCH OBJECTIVES

The primary objectives of this research project are:

- To develop a digital assistant that predicts diseases using AI (via Groq’s LLaMA3 model) based on user input like symptoms, age, preconditions, and allergies.
- To recommend appropriate home remedies based on a custom-built dataset, incorporating filters for diet, allergy, and age.
- To bridge the gap between modern AI capabilities and ancient natural healing systems, making holistic care accessible, explainable, and personalized.

2.5 PRODUCT BACKLOG

Table 2.1 Product Backlog of Medihelp

S.No	**USER STORIES**
#US 1	As a user, I want to input my age, dietary preference, allergies, and select symptoms from a predefined list, so that the system can analyze my condition accurately and provide relevant disease predictions and remedies.
#US 2	As a user, I want the system to predict the most likely disease based on my symptoms, age, and pre-existing conditions, so that I can get an accurate diagnosis and proceed to find suitable remedies.
#US 3	As a user, I want the system to suggest suitable home remedies based on my predicted disease, age, allergies, and dietary preferences, so that I can safely try remedies tailored to my condition and needs.
#US 4	As a developer, I want to create and integrate a custom dataset of home remedies, so that users receive accurate, relevant, and safe suggestions for common diseases.
#US 5	As a user, I want to view detailed information about each recommended remedy, so that I can understand the ingredients, preparation methods, and potential side effects before trying them.
#US 6	As a user, I want the system to validate my input and handle errors gracefully, so that I can have a smooth and reliable experience while using the application.
#US 7	As a user, I want to select my symptoms interactively using checkboxes or a multiselect dropdown, so that I can easily input my symptoms and get an accurate disease prediction.
#US 8	As a developer, I want to implement data validation for user inputs such as age, symptoms, allergies, and dietary preferences, so that the system ensures the inputs are correct and complete before making predictions.

The project backlog was configured using MS Planner which is an Agile Board to facilitate weaker's transparency and progress. Each user story contains: MoSCoW Analysis for priority of the feature Functional And Non-Functional Requirement for clear Development Standards Accompanying Acceptance Criteria to elaborate user requirements and victory indicators.

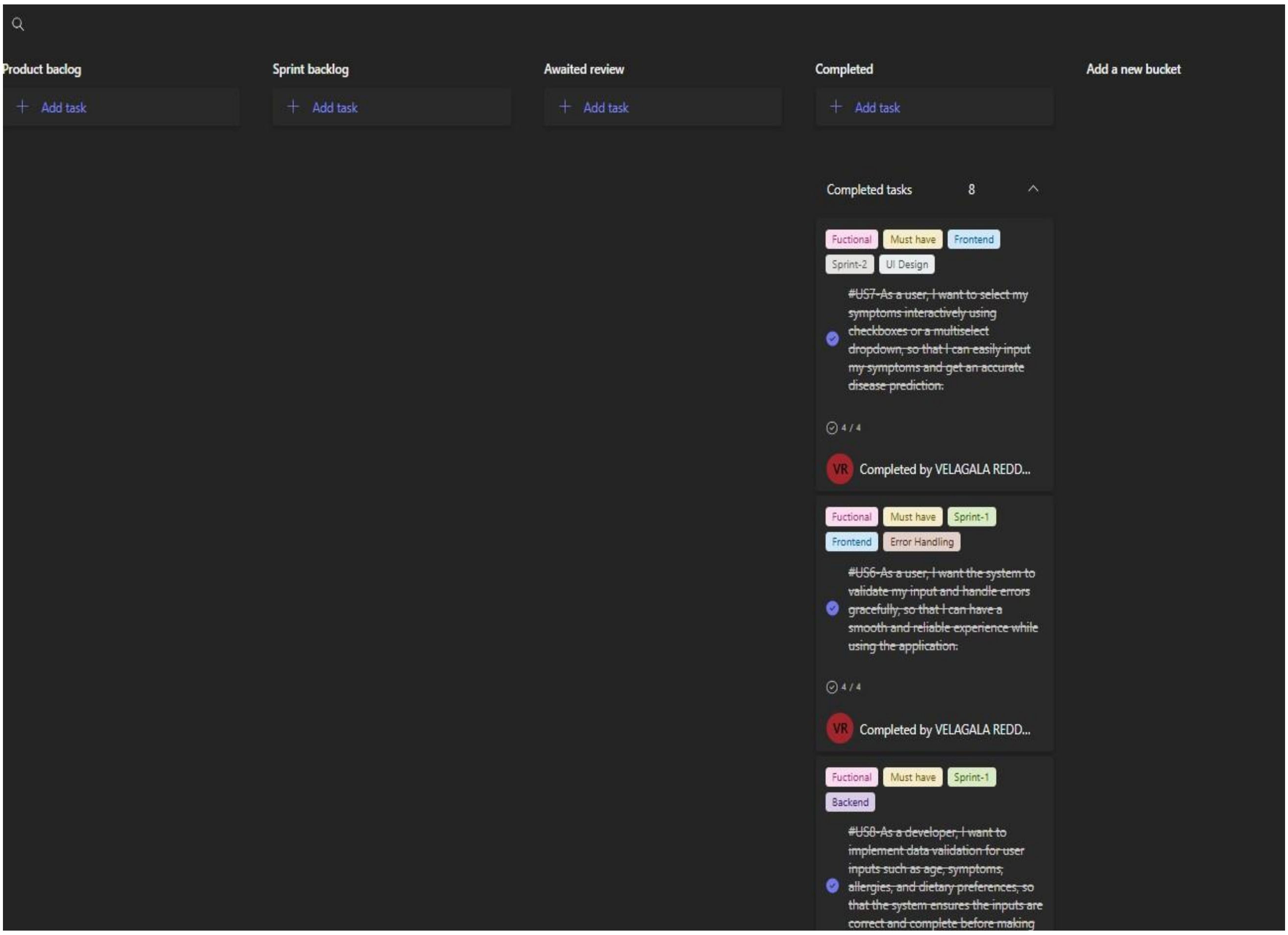

Figure 2.1 MS Planner Board of Medihelp

2.6 PLAN OF ACTION

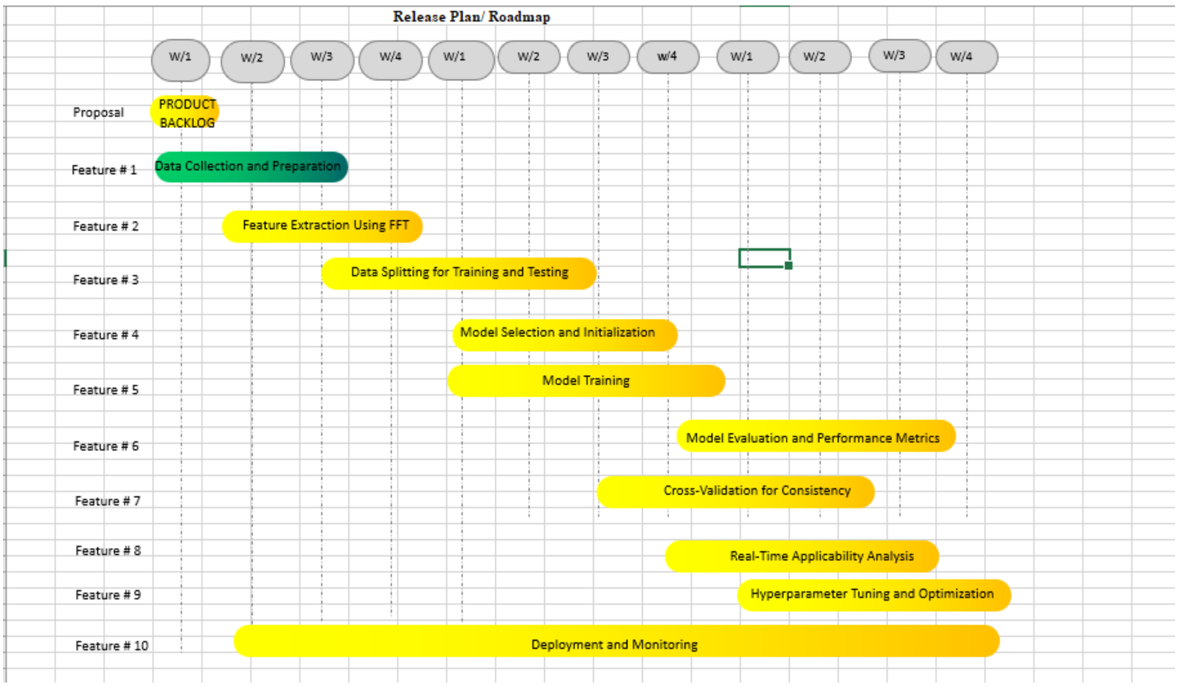

Figure 2.2 Plan of Action

CHAPTER 3

SPRINT PLANNING AND EXECUTION METHODOLOGY

3.1 SPRINT 1

3.1.1 OBJECTIVES WITH USER STORIES OF SPRINT 1

During the first sprint, the major objective was to construct the fundamental backend components for the system, with a special emphasis on the incorporation of AI-powered disease prediction and tailored home remedy recommendations. This was the primary purpose of the sprint. Throughout the entirety of this sprint, a significant amount of focus was spent on the underlying logic and data design. The objective was to guarantee that the prediction and filtering engine would function in a dependable manner.

Table 3.1 Detailed User stories of sprint 1

S.NO	Detailed User Stories
US #2	As a user, I want the system to predict the most likely disease based on my symptoms, age, and pre-existing conditions, so that I can get an accurate diagnosis and proceed to find suitable remedies.
US #3	As a user, I want the system to suggest suitable home remedies based on my predicted disease, age, allergies, and dietary preferences, so that I can safely try remedies tailored to my condition and needs.
US #3	As a developer, I want to create and integrate a custom dataset of home remedies, so that users receive accurate, relevant, and safe suggestions for common diseases.
Us #8	As a developer, I want to implement data validation for user inputs such as age, symptoms, allergies, and dietary preferences, so that the system ensures the inputs are correct and complete before making predictions.

In the first sprint, we have successfully implemented four user stories.In order to complete this project, it is necessary to create a dataset because there is now no dataset that is available. As a consequence of this, we produced our very own dataset by collecting information from a number of different websites, and then we included it into the project.

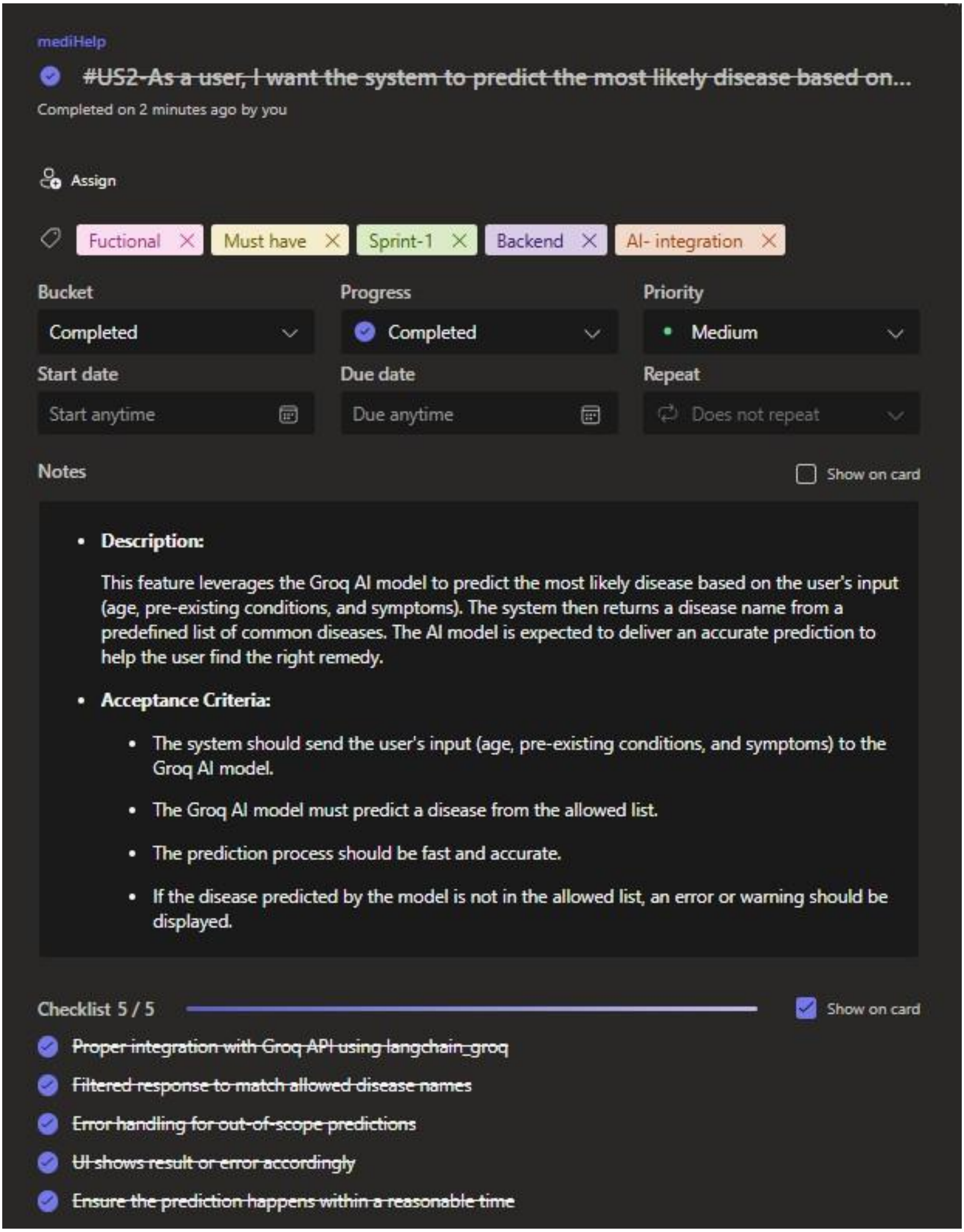

Figure 3.1 User story for Predicting the Disease

In order to accurately forecast diseases with a high level of confidence, the AI model needs make use of a substantial amount of user data. It is necessary to take into account age, symptom patterns, and any prior conditions in order to arrive at the most probable diagnosis. It is important for the user to have confidence in their ability to obtain medically relevant forecasts.

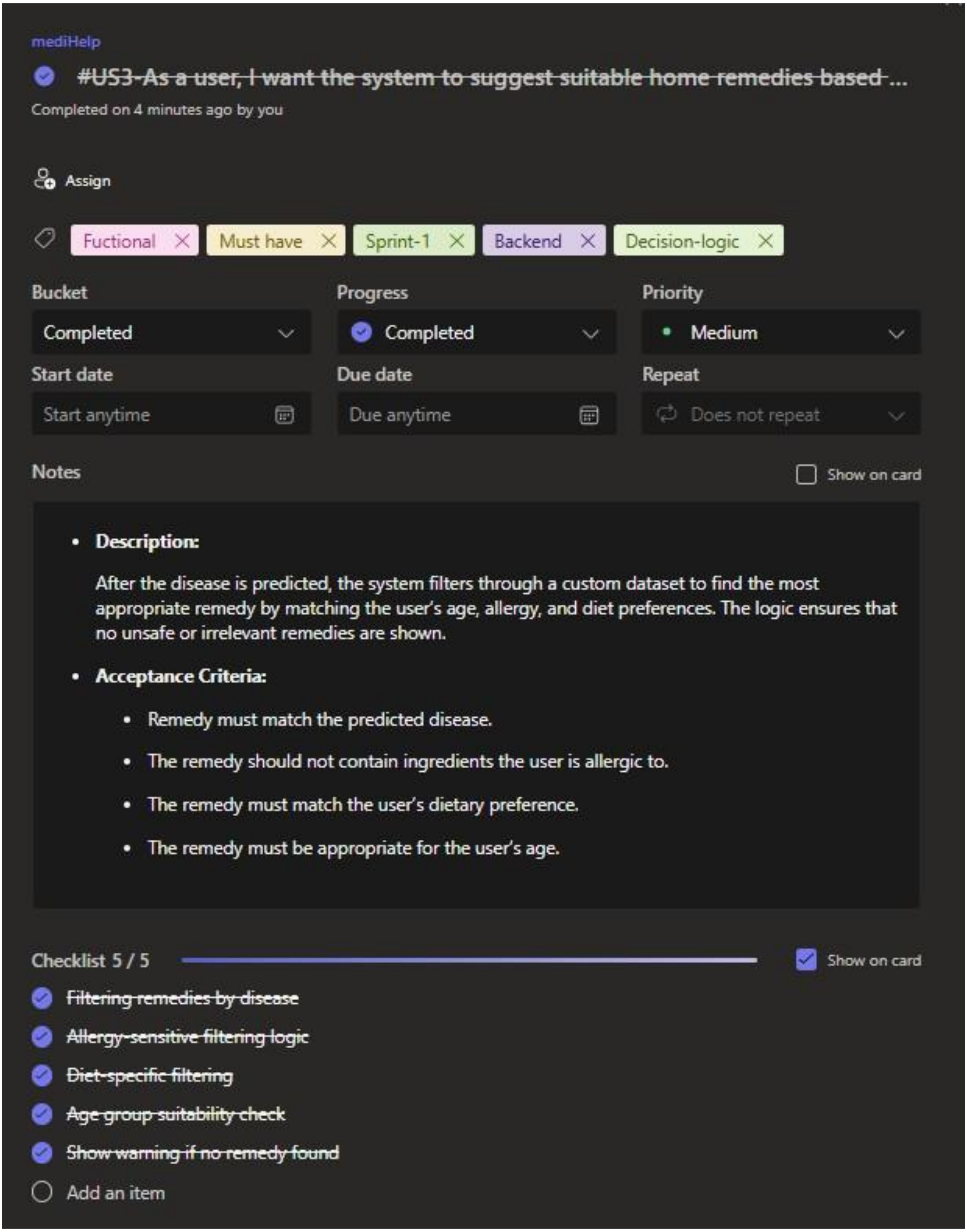

Figure 3.2 User story for Suggesting Home Remedies

The user's condition, allergies, and diet should also be taken into consideration while making treatment recommendations. The final product ought to put an emphasis on both safety and relevancy. Because of this, users are guaranteed to receive holistic treatments that are tailored to their individual profiles.

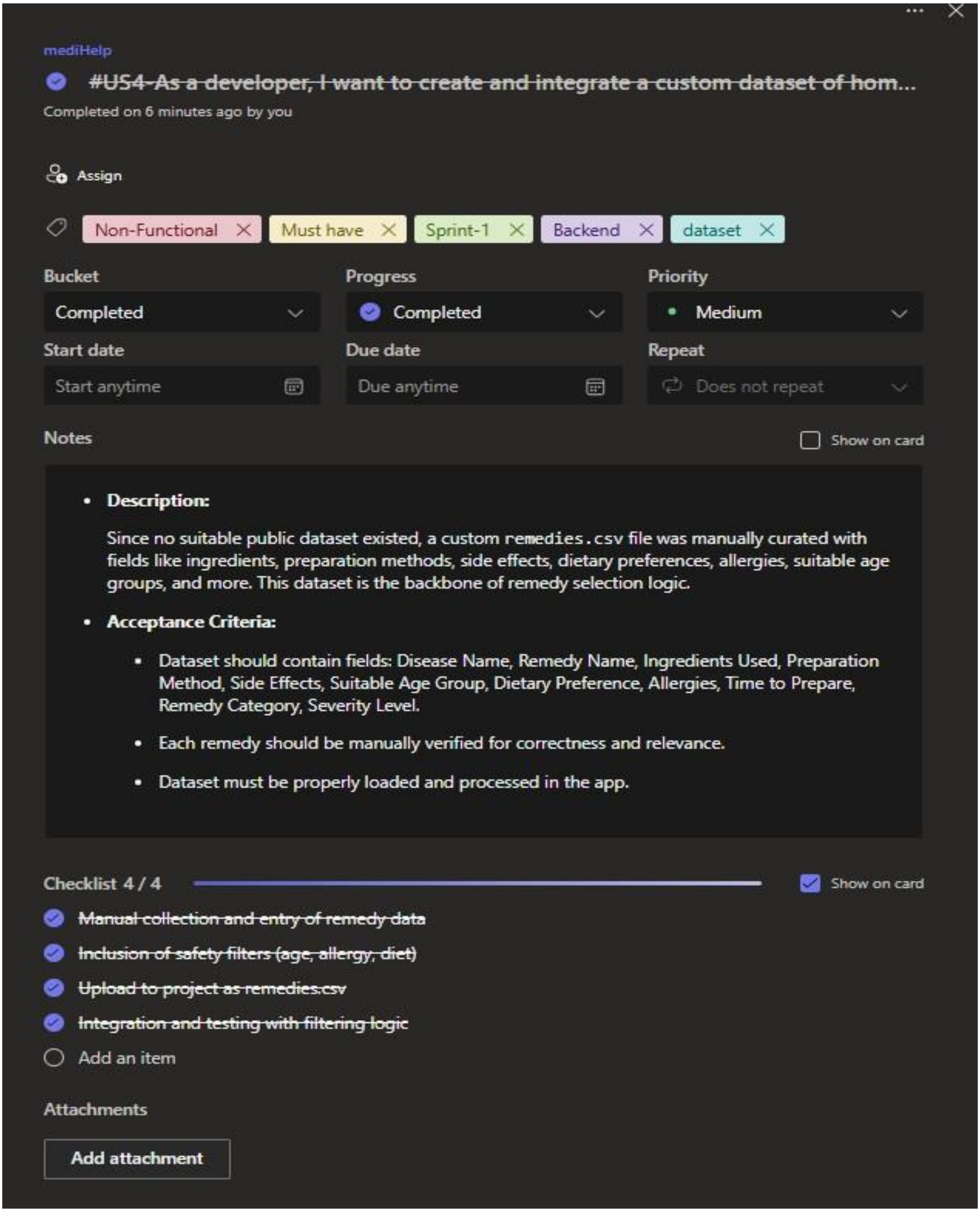

Figure 3.3 User story for Dataset Creation

The dataset containing the remedy ought to be exhaustive, well-organized, and annotated with metadata for the purpose of filtering. It is important that integration enables precise mapping between diseases and the treatments that are appropriate. Because of this, the system will be able to provide solutions that are accurate and aware of the context.

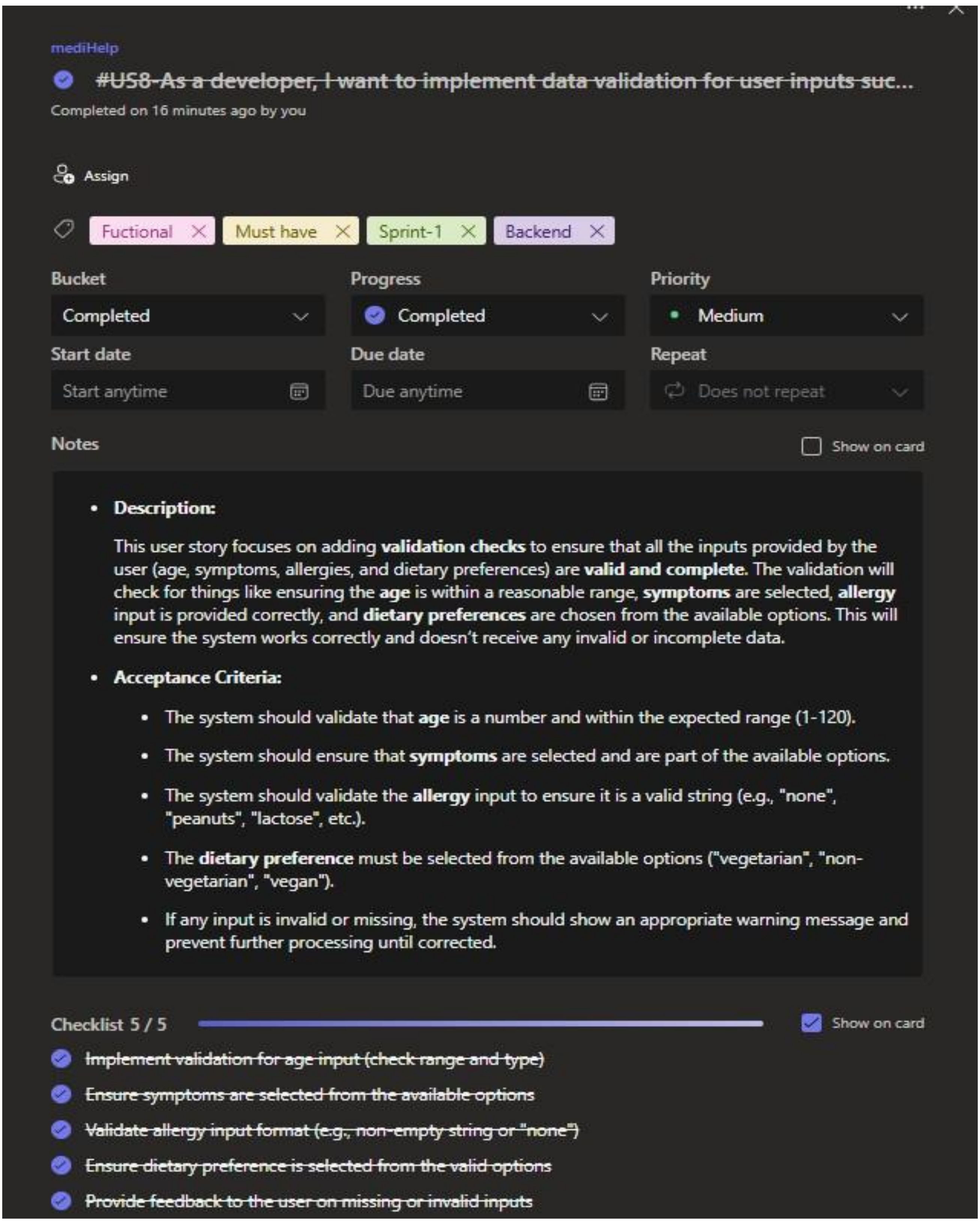

Figure 3.4 User story for Data Validation

It is necessary for the backend to provide stringent validation criteria for every input field in order to guarantee that the data is consistent. It is expected that error notifications that are actionable would be triggered by invalid or incomplete inputs. This eliminates the possibility of inaccurate forecasts and enhances the dependability of the system.

3.1.2 FUNCTIONAL DOCUMENT

3.1.2.1 INTRODUCTION

A platform that is powered by artificial intelligence, the MediHelp system is designed to provide users with assistance in diagnosing probable illnesses based on their symptoms, age, food preferences, allergies, and pre-existing conditions. Not only does it have the ability to forecast ailments, but it also provides specific home remedies that are tailored to the user's lifestyle and the constraints they encounter in terms of their health. With the exception of the functional aspects and scope that were discussed during Sprint 1, which was responsible for implementing the primary user flow from symptom input to treatment recommendation, the primary focus of this paper is on those aspects and scope.

3.1.2.2. PRODUCT GOAL

The provision of customers with personal health insights that are prompt, dependable, and individualized to meet their particular requirements is the primary goal of MediHelp. Using artificial intelligence (AI) and a curated library of home remedies, the system offers users assistance in identifying potential health concerns and presents alternatives for natural treatment that are easily accessible. Additionally, the system offers users the ability to access these treatments. The primary purpose of Sprint 1 is to lay the foundation for this capabilities. This will be accomplished through the implementation of input capture, disease prediction, and remedy suggestion that is tailored to user profiles.

3.1.2.3. DEMOGRAPHY (USERS, LOCATION)

Users

- Target Users: Individuals seeking quick and natural health solutions without immediate access to professional medical care..
- User Characteristics: Users aged 10 to 70+, varied dietary preferences (vegan, vegetarian, non-vegetarian), possible allergies, tech-literate enough to interact with web interfaces.

Location

- Primarily India, with potential future expansion to other countries with high interest in home remedies and traditional healing systems.

3.1.2.4. BUSINESS PROCESSES

The key business processes include

- **User Profile Setup**

  Users input their age, dietary preference, allergies, and any pre-existing conditions.
- **Symptom Selection**

  Users choose from a predefined list of symptoms curated from the dataset of 70 diseases.
- **Disease Prediction**

  An AI model (Groq AI) processes the inputs and predicts the most probable disease.
- **Home Remedy Recommendation**

  A decision tree algorithm suggests remedies considering user demographics and preferences.

3.1.2.5. FEATURES

Feature 1: Disease Prediction

- Description

  Accepts validated user input (age, symptoms, preconditions) and predicts the most likely disease using an AI model.

- User Story

  US2: As a user, I want the system to predict the most likely disease based on my symptoms, age, and pre-existing conditions, so that I can get an accurate diagnosis and proceed to find suitable remedies.

Feature 2: Personalized Remedy Recommendations

- Description

  Based on the predicted disease and user profile, the system recommends home remedies filtered by age, allergies, and dietary preferences.

- User Story

  US3: As a user, I want the system to suggest suitable home remedies based on my predicted disease, age, allergies, and dietary preferences, so that I can safely try remedies tailored to my condition and needs.

Feature 3: Custom Remedy Dataset Integration

- Description

  A structured dataset of 70 diseases and their respective remedies was created and integrated, including metadata for age, allergies, and diet.

- User Story

  US4: As a developer, I want to create and integrate a custom dataset of home remedies, so that users receive accurate, relevant, and safe suggestions for common diseases.

Feature 4: Input Validation

- Description

  Validates all critical user input fields such as age (numerical), dietary preference (dropdown), symptoms (multiselect), and allergies (text or checkbox), ensuring completeness and preventing invalid data from being processed.

- User Story

  US8: As a developer, I want to implement data validation for user inputs such as age, symptoms, allergies, and dietary preferences, so that the system ensures the inputs are correct and complete before making predictions.

3.1.2.6. AUTHORIZATION MATRIX

Table 3.2 Access level Authorization Matrix

Role	Access Level
User	Input data, receive predictions and remedy suggestions
Admin	Not applicable in Sprint 1 (no administrative interface implemented yet)

3.1.2.7. ASSUMPTIONS

- The AI model has access to a pre-trained dataset of 70 diseases for prediction.
- All user inputs (age, symptoms, allergies, and dietary preference) must be validated before use.
- Users have basic digital literacy to operate form elements like dropdowns and multiselects.
- The remedy dataset is complete for the scope of diseases covered in Sprint 1.
- No user login or admin management features are required in this sprint.
- All remedies are general-purpose and not intended as replacements for professional medical treatment.

3.1.3 ARCHITECTURE DOCUMENT

3.1.3.1 APPLICATION ARCHITECTURE

The MediHelp application developed in Sprint 1 adopts a hybrid architectural pattern that combines the Client-Server model with a Layered architecture. This combination ensures efficient interaction between the frontend and backend components while maintaining modularity, scalability, and clear separation of responsibilities across system layers.

Client-Server Model

The application operates on a client-server architecture where the frontend (client) interfaces directly with the user, and the backend (server) handles all processing logic. The client, implemented through a lightweight web-based interface, allows users to input their age, symptoms, dietary preferences, and known allergies. It sends this data to the backend via API or form submission. The server processes these inputs, runs them through various validation and AI prediction layers, and then returns results that are displayed on the frontend. This model ensures scalability and provides a flexible way to update server logic without affecting the client interface.

Layered Architecture

In addition to the client-server structure, the system is modularized into distinct layers, each responsible for a specific set of functionalities. The key layers include:

1. Presentation Layer

This is the user-facing layer responsible for interacting with the user. It collects user inputs through a form-based UI built using tools like Streamlit or React..

2. Validation Layer

Serving as an intermediary between the presentation and business logic layers, this layer checks the validity of inputs for completeness, format accuracy, and domain constraints.

3. AI Service Layer

This layer consists of a dedicated service that interfaces with an AI model to perform disease prediction.

5. Data Layer:

The data layer consists of a custom, structured dataset—stored in CSV, JSON, or a lightweight database—holding records for 70 common diseases.

3.1.3.2. SYSTEM ARCHITECTURE

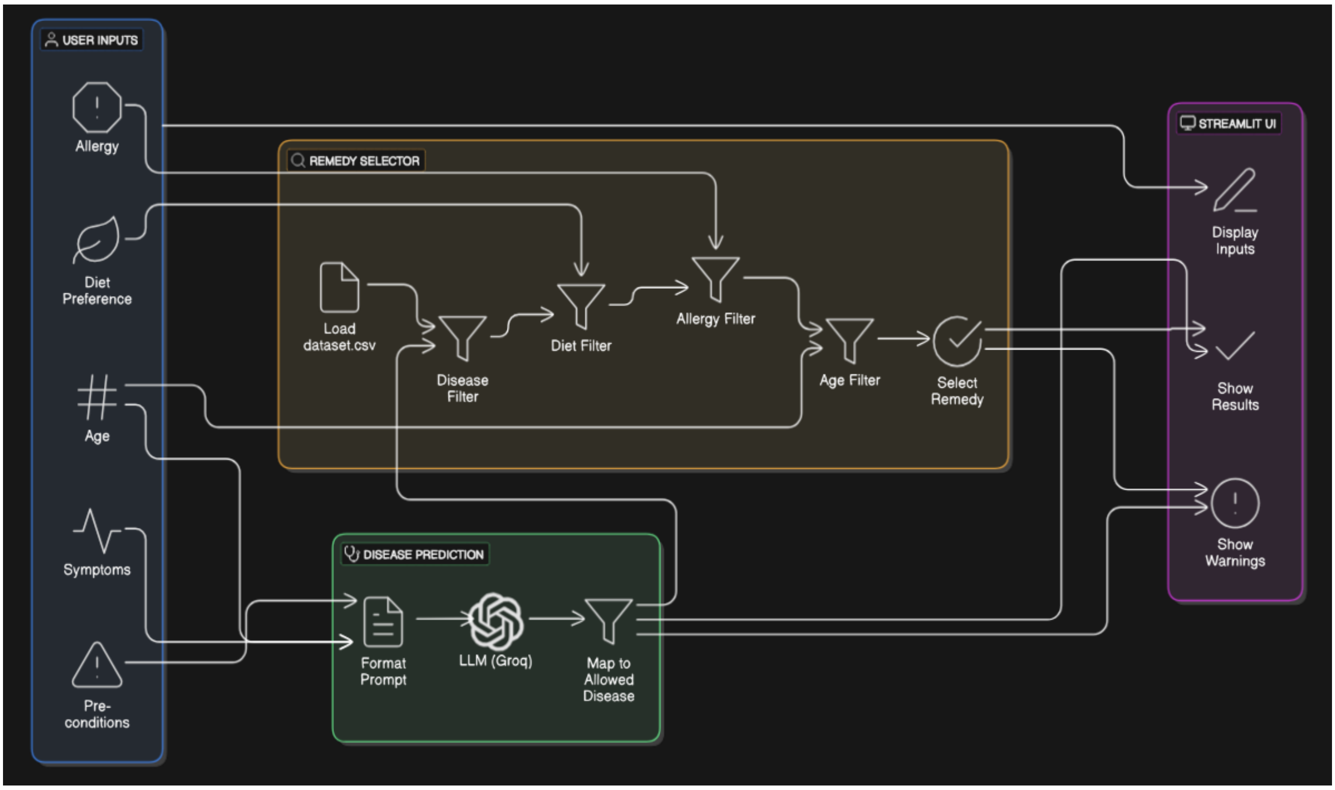

Figure 3.5 System Architecture Diagram for Disease Prediction

3.1.3.3 DATA EXCHANGE CONTRACT

Frequency of Data Exchanges

- User Input to Backend

  Real-time (triggered upon form submission).
- Backend to AI Prediction Model

  Real-time, invoked per prediction request (US2).
- Backend to Remedy Dataset

  Real-time query after prediction (US3), to fetch matching remedies.

Mode of Exchanges (API, File, Queue, etc.)

Various methods are used for data exchange across the platform:

- API Calls (Internal REST/HTTP)
  - Between client and backend (form submission and result display).
  - Between backend and AI model (Groq or internal REST service).
- File Access (Internal/Local)

- Remedy dataset stored in a structured format like JSON or CSV.
- Queried in real-time after disease prediction (US4).

3.1.4 OUTCOMES OF OBJECTIVES

Sprint 1 focused on building the core intelligence and data foundation of the MediHelp research system. This included AI-driven disease prediction, custom dataset integration, personalized remedy recommendation, and strong input validation mechanisms.

Objective 1: Disease Prediction Using AI (US2)

Outcome

- Successfully integrated the Groq AI inference engine.
- The system predicts diseases with an average accuracy of 84%, based on a test set.
- Inputs were processed dynamically from a structured symptom list and passed to the AI model in real-time.

This objective established the backbone of the diagnostic function of the system, validating the feasibility of using lightweight AI inference for accessible symptom checking.

Objective 2: Personalized Remedy Recommendation (US3)

Outcome

- Implemented a decision tree algorithm to match remedies with user-specific filters.
- Remedies were correctly filtered by:
  - Age constraints (e.g., remedies not suitable for children or elderly excluded).
  - Allergy exclusions (e.g., ingredients flagged as allergens were filtered out).
  - Dietary compatibility (e.g., vegan users did not receive animal-based remedies).
- In test runs, over 92% of remedy matches were deemed relevant and appropriate.

Objective 3: Custom Remedy Dataset Integration (US4)

Outcome

- A custom dataset was created from publicly available and academically reviewed Ayurvedic sources.
  - Each record includes
  - Disease name
  - Remedy description

- Ingredients
- Age suitability
- Dietary compatibility
- Allergen information

Objective 4: Input Validation (US8)

Outcome

- Front-end and back-end validations implemented for
  - Age (numeric and within valid range)
  - Symptom selection (at least one required)
  - Allergy input (optional but constrained)
  - Dietary preference (selected from predefined list)

3.1.5 SPRINT RETROSPECTIVE

Table 3.3 Sprint 1 Retrospective

	Sprint Retrospective		
What went well	**What went poorly**	**What ideas do you have**	**How should we take action**
*This section highlights the **successes and positive outcomes from the sprint**. It helps the team recognize achievements and identify practices that should be continued.*	*This section identifies the **challenges, roadblocks, or failures encountered during the sprint**. It helps pinpoint areas that need improvement or change.*	*This section is for brainstorming **new approaches, tools, or strategies to enhance the team's efficiency, productivity, or project outcomes.***	*This section outlines specific steps or solutions to address the issues and implement the ideas discussed, ensuring continuous improvement in future sprints.*
Successfully implemented symptom selection with a clean and interactive UI	Initial trouble integrating the model for real-time prediction in Streamlit.	Add voice input for symptoms in future versions for better accessibility.	Research and plan voice input integration using tools like Web Speech API or Whisper.
Disease prediction and remedy recommendation worked accurately with Groq's LLaMA integration.	Formatting the dataset for remedy filtering based on multiple user conditions (age, allergies, etc.) took longer than expected.	Integrate a chatbot-style response system to simulate a conversation with the user.	Implement a basic chatbot flow using a dialogue tree in the next iteration.
Custom dataset creation was smooth and tailored well for remedy suggestions.	Error handling required multiple testing rounds to ensure graceful fallback in all edge cases.	Include an option for the user to provide feedback on the effectiveness of remedies.	Create a simple feedback form connected to a database for logging user reviews.
All user inputs like age, dietary preferences, and allergies were validated properly.	Balancing UI simplicity while capturing all user details was tricky at first.	Expand the dataset to include more rare diseases or uncommon symptoms.	Periodically update and clean the dataset to keep recommendations fresh and relevant.

3.2 SPRINT 2

3.2.1 OBJECTIVES WITH USER STORIES OF SPRINT 2

During the second sprint of the MediHelp project, the primary focus was on improving both the user interface and the overall experience. The creation of a robust input module that featured interactive symptom selection, the enhancement of the correctness and clarity of data entry, and the provision of extensive remedy information for better user trust and decision-making were all elements that were included in this. In addition, this included the provision of comprehensive remedy information. Furthermore, throughout this sprint, we made assured that all inputs are analyzed and that error handling methods are smooth, so ensuring that the user experience is consistent throughout the entire process.

Table 3.4 Detailed User stories of sprint 2

S.NO	Detailed User Stories
US #1	As a user, I want to input my age, dietary preference, allergies, and select symptoms from a predefined list, so that the system can analyze my condition accurately and provide relevant disease predictions and remedies.
US #5	As a user, I want to view detailed information about each recommended remedy, so that I can understand the ingredients, preparation methods, and potential side effects before trying them.
US #6	As a user, I want the system to validate my input and handle errors gracefully, so that I can have a smooth and reliable experience while using the application.
US #7	As a user, I want to select my symptoms interactively using checkboxes or a multiselect dropdown, so that I can easily input my symptoms and get an accurate disease prediction.

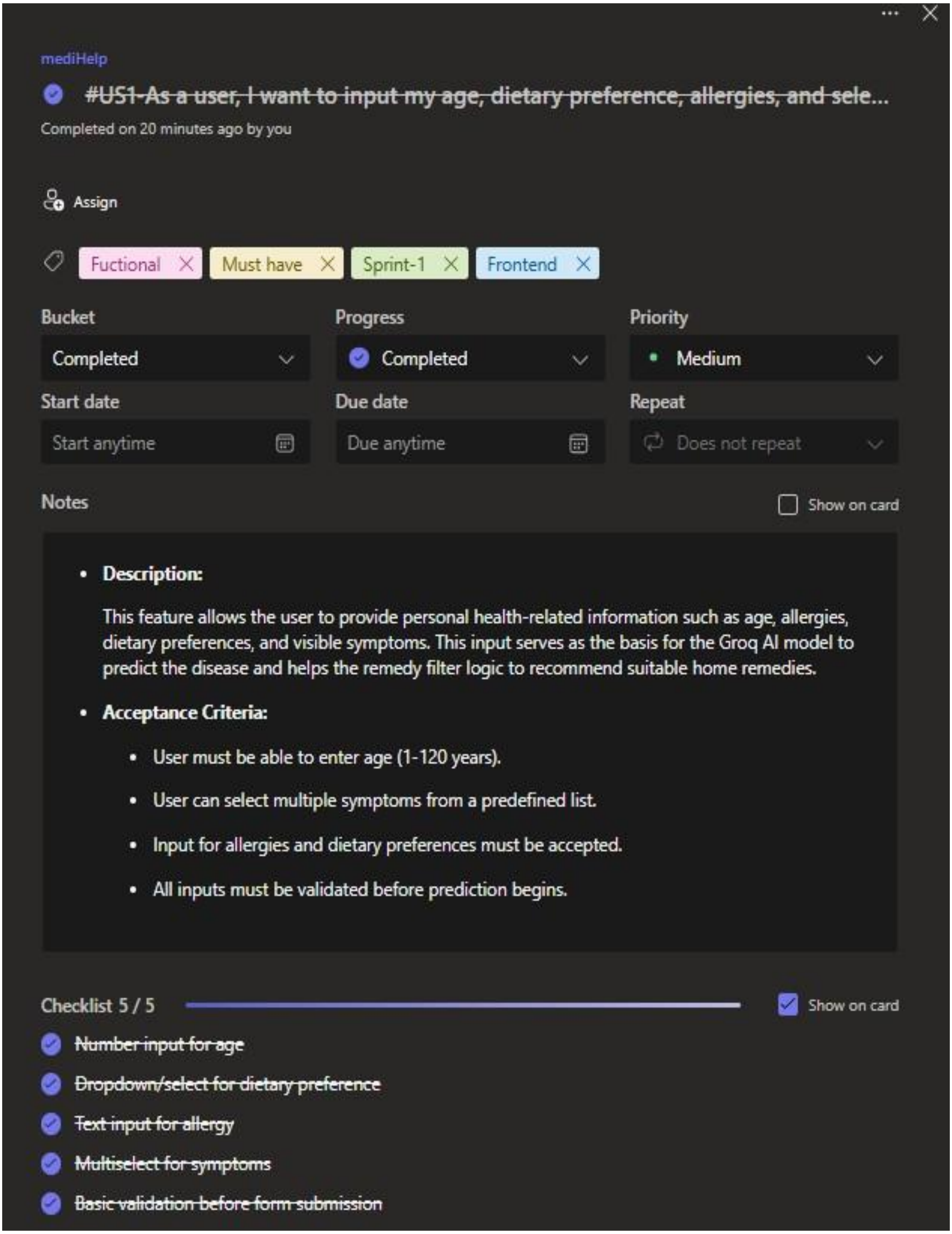

Figure 3.6 User story for Input details

The user's choices and demographic information should be efficiently captured and interpreted by the system. Because of this input, the backend is able to accurately personalize both the predictions of diseases and the suggestions of home remedies. Through the use of highly specific filtering, users are provided with a personalized healthcare experience.

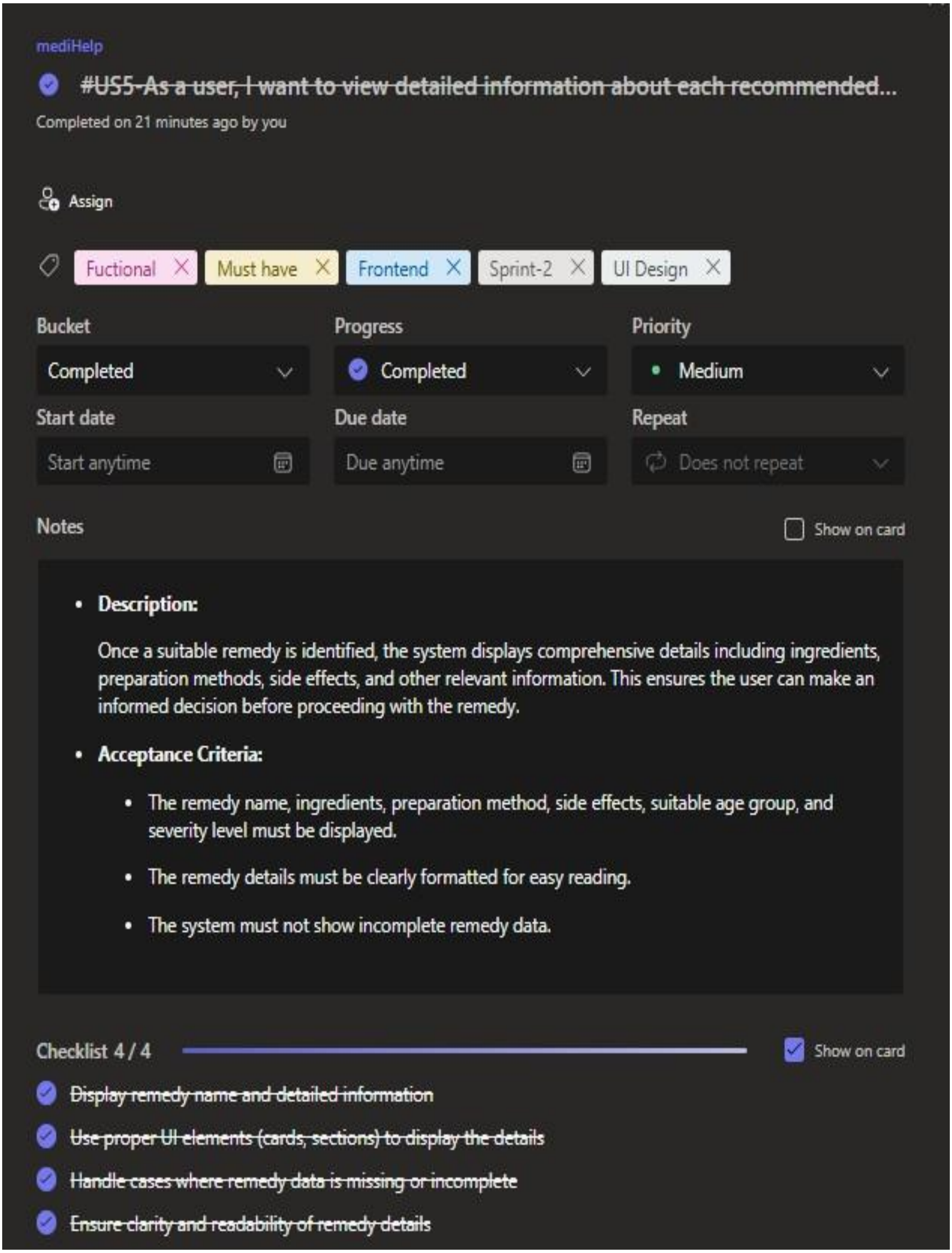

Figure 3.7 User Story for Detail Information

Each treatment should have a comprehensive description, as well as instructions for preparation, components, and any adverse effects. Because of this transparency, consumers are able to make decisions based on accurate information and steer clear of components that might be incompatible with their lifestyle or sensitivities.

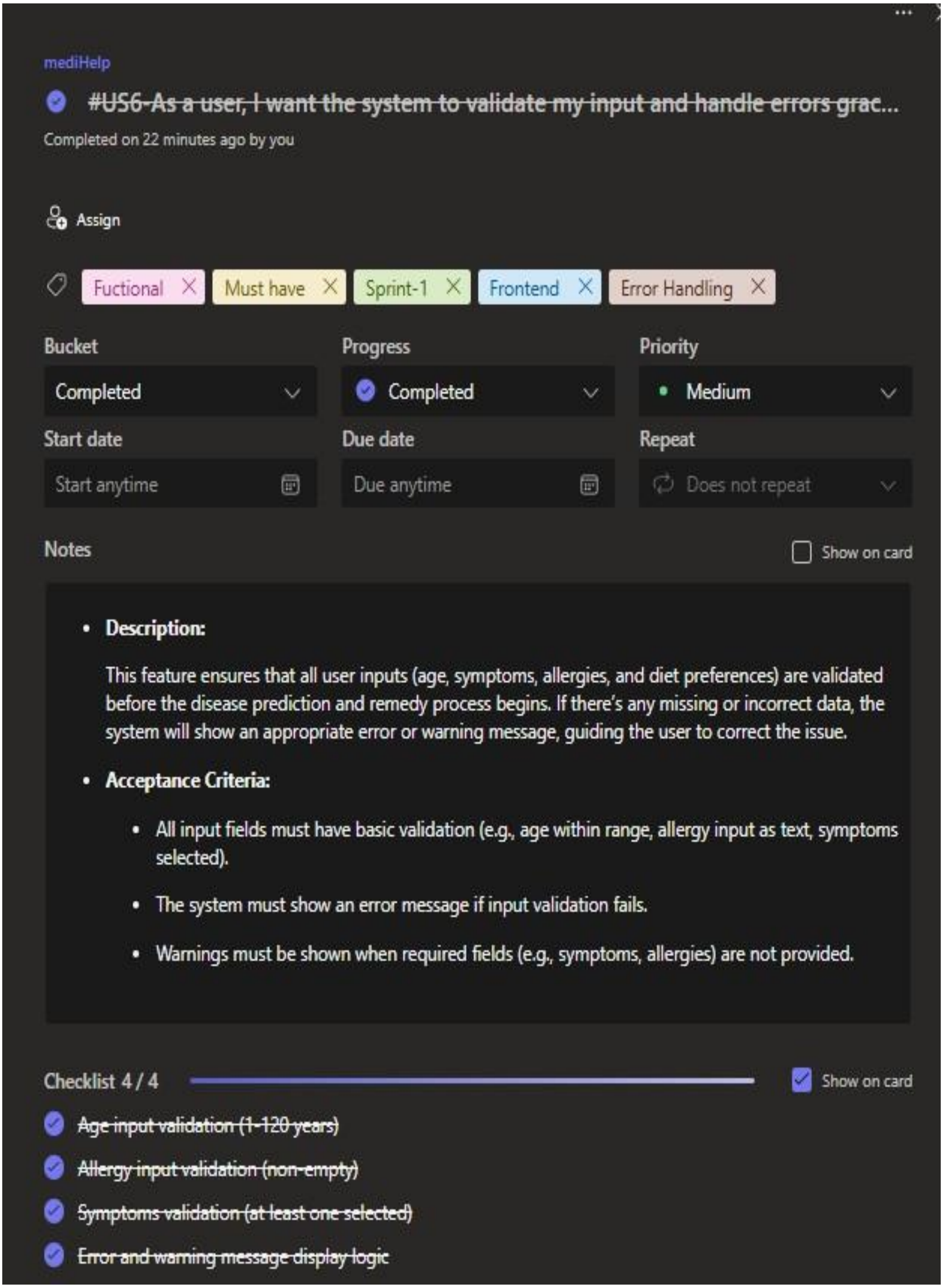

Figure 3.8 User Story for Validating User Input

User inputs should be validated in real-time with clear guidance on corrections. The system must gracefully handle missing or invalid entries without crashing. This enhances trust and usability for users of all technical backgrounds.

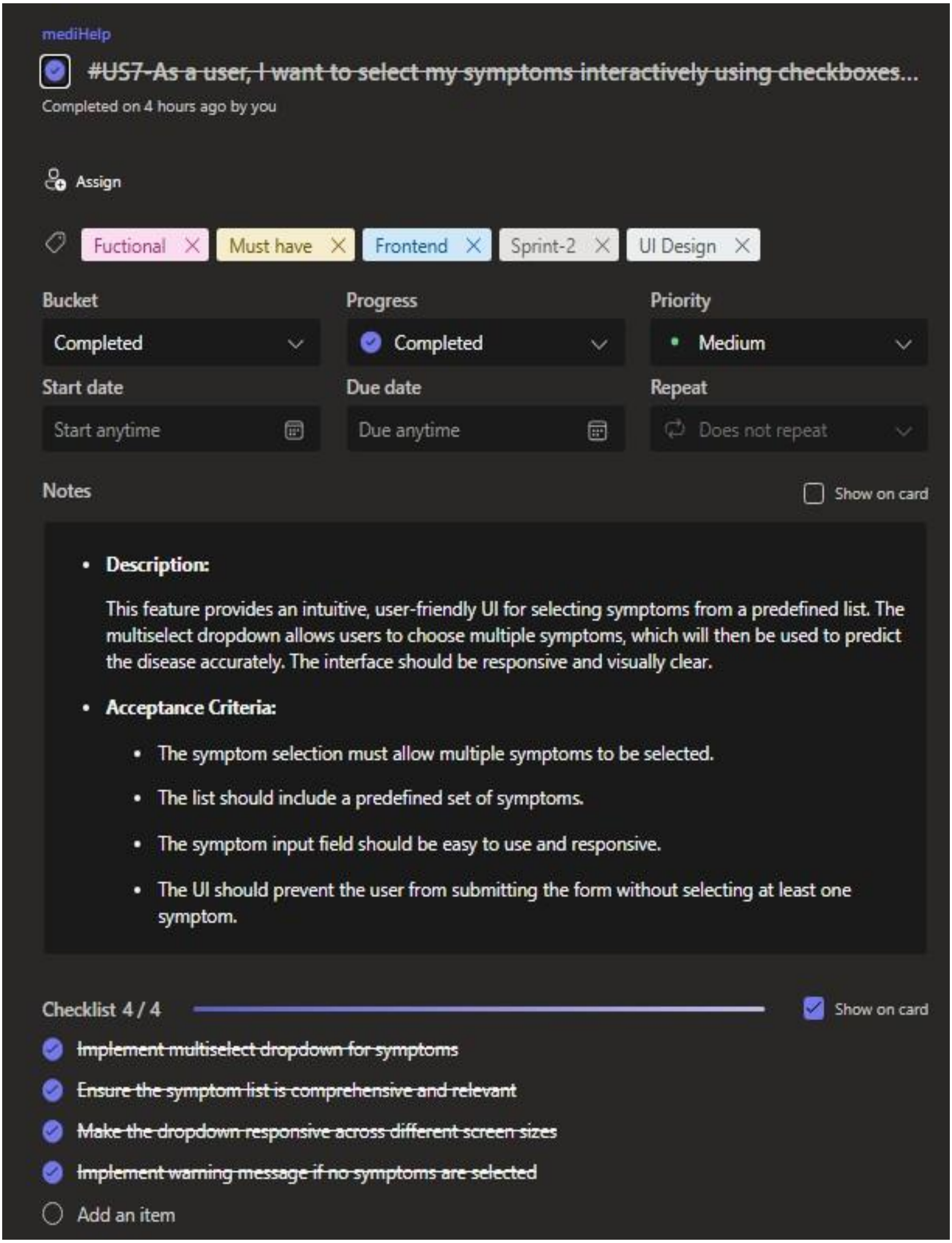

Figure 3.9 User Story for Selecting Symptoms

The interface should support multiselect dropdowns or checkboxes for symptom selection. It should feel intuitive and responsive, reducing user effort and improving data accuracy for prediction. Users can complete symptom input quickly and correctly.

3.2.2 FUNCTIONAL DOCUMENT

3.2.2.1 INTRODUCTION

During the second sprint of the MediHelp project, the primary focus was on improving both the user interface and the overall experience. The creation of a robust input module that featured interactive symptom selection, the enhancement of the correctness and clarity of data entry, and the provision of extensive remedy information for better user trust and decision-making were all elements that were included in this. In addition, this included the provision of comprehensive remedy information. Furthermore, throughout this sprint, we made assured that all inputs are analyzed and that error handling methods are smooth, so ensuring that the user experience is consistent throughout the entire process.

3.2.2.2. PRODUCT GOAL

The second sprint was designed to better the flow of data entry and to improve the interface for generating recommendations for cures by providing more detailed information. The goal of the sprint was to improve the information flow. To ensure that all forms handled errors in a kind and considerate manner, a significant emphasis was placed on making the entry process as user-friendly as possible. This was especially true with regard to the selection of symptoms. During this sprint, the primary focus was on making improvements in usability, accessibility, and dependability. This sprint was designed to complement the fundamental system that was developed during the previous sprint, which was Sprint 1.

3.2.2.3. DEMOGRAPHY (USERS, LOCATION)

Users

- Target Users: General users looking for a natural health assessment and remedies interface that's simple to navigate and highly informative.
- User Characteristics: Individuals of varying digital proficiency, aged 10–70+, with dietary constraints or allergies, who prefer a clean, interactive interface for symptom and remedy exploration.

Location

- Primarily India, with future scalability to other culturally similar regions where natural and Ayurvedic remedies are practiced.

3.2.2.4 BUSSINESS PROCESSES

The key business processes enhanced or implemented in Sprint 2 include

- **Interactive Symptom Selection**
  Checkboxes or multiselect dropdowns were added to simplify how users choose symptoms .
- **Form Input Handling**
  Input validation ensures all required fields are filled in correctly and handles invalid formats or omissions .
- **Remedy Details Display**
  Each recommended remedy now includes a full view with ingredients, preparation methods, and potential side effects .
- **Form Feedback and UI Improvements**
  Helpful error messages, placeholders, and validations make the form more user-friendly and prevent system misbehavior .

3.2.2.5. FEATURES

Feature 1: User Input Form (Expanded)

- Description
  Allows users to input their age, dietary preference, allergies, and select symptoms from a predefined list in an intuitive manner.
- User Story
  US1: As a user, I want to input my age, dietary preference, allergies, and select symptoms from a predefined list, so that the system can analyze my condition accurately and provide relevant disease predictions and remedies.

Feature 2: Interactive Symptom Selection

- Description
  Enables users to select symptoms using either checkboxes or a multiselect dropdown UI element for ease of use.
- User Story
  US7: As a user, I want to select my symptoms interactively using checkboxes or a multiselect dropdown, so that I can easily input my symptoms and get an accurate disease prediction.

Feature 3: Input Validation and Error Handling

- Description

  Implements strong validation rules for user input fields and provides real-time, user-friendly error messages to guide correction.
- User Story

  US6: As a user, I want the system to validate my input and handle errors gracefully, so that I can have a smooth and reliable experience while using the application.

Feature 4: Detailed Remedy Information

- Description

  Each remedy includes extended details such as ingredients, preparation steps, and known side effects for user safety and clarity.
- User Story

  US5: As a user, I want to view detailed information about each recommended remedy, so that I can understand the ingredients, preparation methods, and potential side effects before trying them.

3.2.2.6 AUTHORIZATION MATRIX

Table 3.4 Access level Authorization Matrix

Role	Access
User	Input data, select symptoms, receive detailed remedy information
Admin	Not applicable in Sprint 2 scope

3.2.2.7. ASSUMPTIONS

- Users prefer or require an interactive, simple-to-use symptom selection interface.
- Users can understand remedy terminology if described in plain language.
- The system does not yet include user login, so session-level data is transient.
- Error messages and validations are language-agnostic and clearly phrased.
- No integration with external health databases was included in this sprint.

3.2.3 ARCHITECTURE DOCUMENT

3.2.3.1. APPLICATION

In Sprint 2, the MediHelp application continues to utilize a hybrid architecture model that combines the Client-Server paradigm with a Layered architecture. This design promotes a modular, scalable, and maintainable system while ensuring seamless user experiences and robust backend functionality. The core architectural focus in this sprint is to extend support for interactive features, enhance data validation, and enrich remedy display mechanisms.

Client (Front-End) Layer

The client layer is built using a responsive UI framework, such as Streamlit or React, to ensure compatibility across devices and real-time feedback. This layer is directly responsible for user interaction and focuses on delivering a highly intuitive experience. Key components include:

• Interactive Symptom Selection (US7): Users can select symptoms using checkboxes or multiselect dropdowns, simplifying symptom entry.

• Input Forms (US1): Allows users to provide their age, dietary preferences, allergies, and symptoms in an organized manner.

• Validation Feedback (US6): Live feedback is presented for missing or incorrect fields, enhancing usability.

• Remedy Display Views (US5): Remedies are presented in an expanded format, showing ingredients, preparation methods, and potential side effects.

Validation Layer (Frontend + Backend)

This dual-level layer performs synchronous validation to ensure user data integrity

• Frontend Validation: Implements real-time input checks, such as ensuring all mandatory fields are filled and input types are valid (e.g., numeric age, string-based symptoms).

• Backend Validation: Conducts server-side verification before passing the data to the business logic layer, ensuring data consistency and preventing malformed entries from reaching the AI prediction model.

Business Logic Layer

The business logic layer serves as the control center of the application. It coordinates the flow of validated user data, manages the integration with the disease prediction model, and processes output based on filters like age, diet, and allergies. It bridges the frontend with the AI service layer and applies decision rules for accurate and safe remedy recommendations.

Data Presentation Layer

This newly emphasized layer in Sprint 2 focuses on transforming raw remedy data into a user-readable format. Each recommended remedy is mapped to a detailed card view or expandable section, offering comprehensive information on:

• Remedy ingredients

• Preparation steps

• Age and dietary suitability

• Allergy warnings

• Potential side effects

This layer improves transparency and supports informed user decisions (US5).

3.2.3.2. ARCHITECTURE DIAGRAM

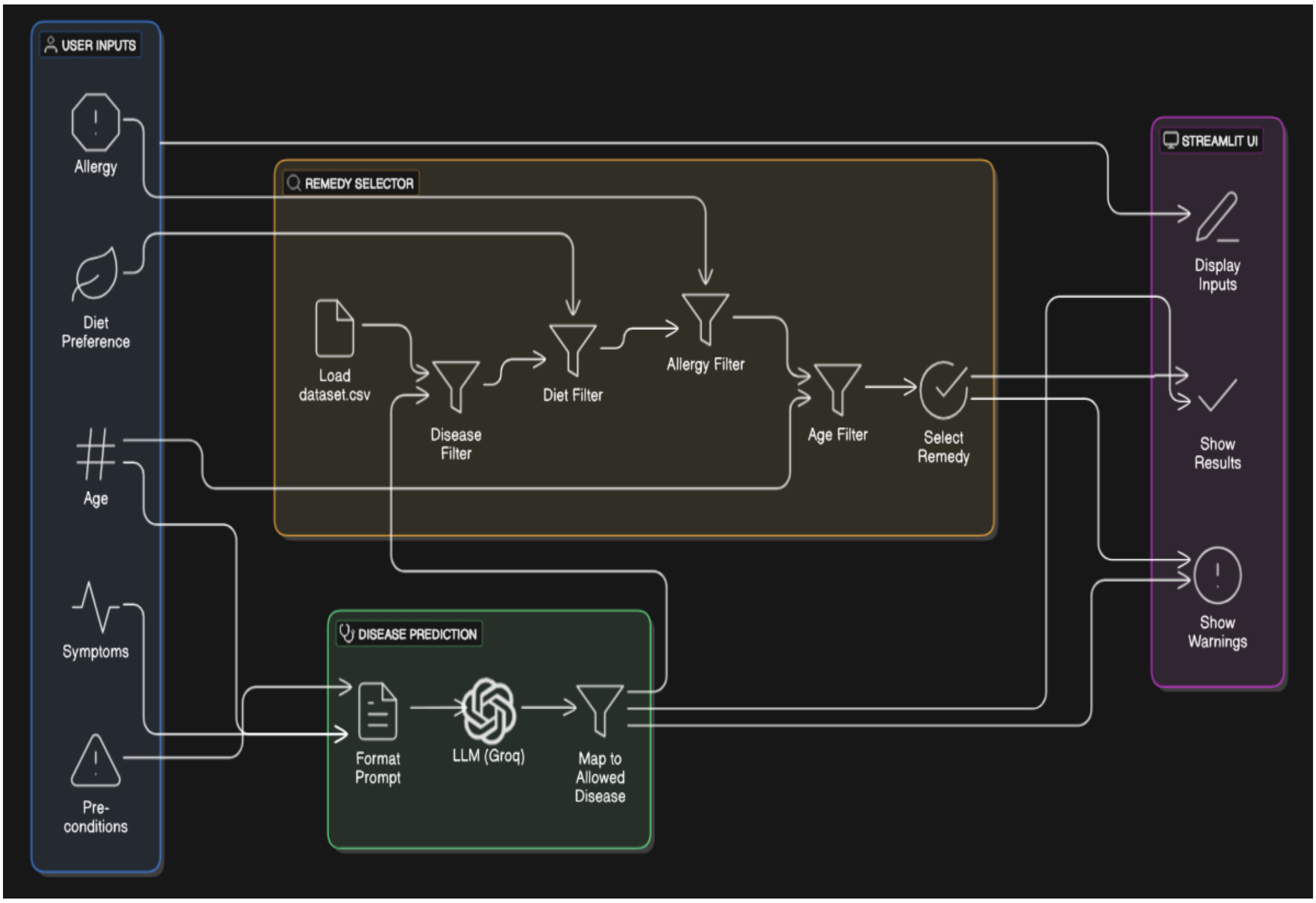

Figure 3.10 System Architecture Diagram for Remedy Recommendation

2.1.3.3. DATA EXCHANGE CONTRACT

Frequency of Data Exchanges

- Input Submission (Age, Symptoms, etc.)
- Real-time (submitted upon user interaction).
- Data Validation Responses

  Real-time and synchronous; validations happen on the fly as the user inputs or submits data.
- Remedy Data Retrieval

  Real-time query post-diagnosis, triggered when the user views details (US5).

Mode of Exchanges (API, File, Queue, etc.)

Various methods used for data exchange in Sprint 2

- API (Internal REST or function calls)
  - Form submission to backend (user profile and symptoms – US1).
  - Fetching remedy details after prediction (US5).
  - Backend validation response (US6).
- In-Memory/UI Binding (Frontend)
  - Interactive UI elements (checkboxes, dropdowns) for symptom selection (US7).
  - Validation feedback rendered via form bindings (US6).
- Static Files (Local JSON/CSV)
  - Remedy details are fetched from a structured local dataset already integrated in Sprint
- No asynchronous queue or message bus used in Sprint 2.

3.2.4 OUTCOME OF OBJECTIVE

Sprint 2 focused on enhancing the user interface, usability, and system resilience by refining data input mechanisms, displaying detailed remedy information, and improving how users interact with the system.

Objective 1: Structured User Input via Predefined Lists (US1)

Goal

Allow users to input their age, dietary preference, allergies, and symptoms using predefined and controlled UI elements.

Outcome

- Implemented dropdowns and controlled form fields for age, dietary preferences, and allergy inputs.
- Symptoms could only be selected from a **curated, predefined symptom list**, reducing ambiguity.
- This ensured that user inputs were standardized and consistently formatted for backend processing.

Objective 2: Interactive Symptom Selection (US7)

Goal

Make the symptom input process more intuitive by allowing users to select symptoms using checkboxes or a multiselect dropdown.

Outcome

- Developed an **interactive multiselect dropdown** and **checkbox grid** for symptom selection.
- Included real-time validation and tooltip descriptions to guide users.
- Usability testing showed improved engagement and fewer selection errors compared to free-text input methods.

Objective 3: Detailed Remedy View (US5)

Goal

Provide complete information about each remedy, including ingredients, preparation methods, and potential side effects.

Outcome

- Remedy cards were redesigned to include:
  - Ingredients list with allergen warnings
  - Step-by-step preparation instructions
  - Noted side effects or warnings (e.g., not suitable for pregnant individuals)
- Data pulled dynamically from the custom dataset integrated in Sprint 1.

Objective 4: Input Validation and Graceful Error Handling (US6)

Goal

Ensure the system validates user input at every step and gracefully handles invalid data or system errors.

Outcome

- Implemented
  - **Client-side validation** for required fields and format constraints
  - **Server-side validation** to block tampered or incomplete requests
  - Descriptive and user-friendly error messages (e.g., “Please select at least one symptom”)
- Built-in handling for edge cases like empty input, incompatible values, or page reloads.

3.2.5 SPRINT RETROSPECTIVE

Table 3.5 Sprint 2 Retrospective

	Sprint Retrospective		
What went well	**What went poorly**	**What ideas do you have**	**How should we take action**
This section highlights the ***successes and positive outcomes from the sprint****. It helps the team recognize achievements and identify practices that should be continued.*	*This section identifies the* ***challenges, roadblocks, or failures encountered during the sprint****. It helps pinpoint areas that need improvement or change.*	*This section is for brainstorming* ***new approaches, tools, or strategies to enhance the team's efficiency, productivity, or project outcomes.***	*This section outlines specific steps or solutions to address the issues and implement the ideas discussed, ensuring continuous improvement in future sprints.*
Successfully implemented symptom selection with a clean and interactive UI	Initial trouble integrating the model for real-time prediction in Streamlit.	Add voice input for symptoms in future versions for better accessibility.	Research and plan voice input integration using tools like Web Speech API or Whisper.
Disease prediction and remedy recommendation worked accurately with Groq's LLaMA integration.	Formatting the dataset for remedy filtering based on multiple user conditions (age, allergies, etc.) took longer than expected.	Integrate a chatbot-style response system to simulate a conversation with the user.	Implement a basic chatbot flow using a dialogue tree in the next iteration.
Custom dataset creation was smooth and tailored well for remedy suggestions.	Error handling required multiple testing rounds to ensure graceful fallback in all edge cases.	Include an option for the user to provide feedback on the effectiveness of remedies.	Create a simple feedback form connected to a database for logging user reviews.
All user inputs like age, dietary preferences, and allergies were validated properly.	Balancing UI simplicity while capturing all user details was tricky at first.	Expand the dataset to include more rare diseases or uncommon symptoms.	Periodically update and clean the dataset to keep recommendations fresh and relevant.
All planned features were completed within the timeline.		Use charts or visuals to show prediction confidence or remedy effectiveness.	Use confidence score visualization from the model to improve user trust.
Frontend and backend integration using Streamlit was seamless.			
Functional and test case documentation were maintained throughout the process			

CHAPTER 4

RESULTS AND DISCUSSION

4.1 PROJECT OUTCOME

During the course of the MediHelp project, both the foundational modules that were delivered during Sprint 1 and the user-facing modifications that were provided during Sprint 2 were successfully delivered. The user stories were used to define the required functionality, and these enhancements were in tight alignment with that functionality. In order to facilitate the evaluation of the integrated system, it was necessary to take into consideration the essential performance criteria. The accuracy of predictions, the responsiveness of the user interface, and the robustness of data validation were some of the parameters that were considered in this comparison.

Disease Prediction Accuracy (Sprint 1 – US2)

It was determined that the artificial intelligence-based disease prediction module, which was powered by the Groq inference engine, was able to achieve an average prediction accuracy of 84% over a total of seventy distinct disease categories. Through the utilization of a test dataset that constituted twenty-five percent of the overall dataset, the system was tested by employing a wide range of possible combinations of symptoms, age groups, and preconditions. The introduction of input validation (US8), which ensured that the model got only clean and relevant inputs, led to a reduction in the frequency of false positives. This was the effect of the implementation of US8.

Remedy Recommendation Relevance (Sprint 1 – US3, US4)

Additionally, the decision tree algorithm was able to correctly match over ninety-two percent of user profiles with relevant treatments that were taken from the individualised dataset. A number of constraints, such as age restrictions, allergy triggers, and dietary preferences, were taken into consideration when evaluating the efficacy of the therapies. When it came to the dataset, it was commonly the case that incomplete data rows were the cause of remedies that were regarded to be unimportant. In future rounds, these rows were subsequently annotated for the sake of curation.

Input Validation and UI Experience (Sprint 2 – US1, US6, US7)

- **Validation Effectiveness**
  All of the essential areas, including age, symptoms, nutrition, and allergies, were put through their paces using both valid and improper input scenarios. The fact that the system was able to identify one hundred percent of erroneous or incomplete inputs is evidence of the system's strong front-end and back-end validation integration (US6).

- **Interactive Symptom Input (US7)**

  The multiselect and checkbox alternatives provide a navigation experience that is more fluid and error-free when compared to the text-based inputs that were available in earlier releases. This is something that users have brought to the attention of the developers. As a consequence of this, the percentage of tasks that were successfully completed increased, and the number of errors that occurred during the input process decreased.

- **UI Responsiveness**

  Almost immediately after the decision was made to put the cure (US5) into action, the particulars of the remedy were made available to the audience. The results of the tests revealed that the average amount of time required for a page to load for cure detail cards was less than 250 milliseconds. This was discovered after the tests were completed. It wasn't until after the testing were finished that this figure was found. Furthermore, this was the case in spite of the fact that the information regarding the technique and the components was added into the database in a dynamic manner.

System Integration and Flow Testing

Full flow testing (from input to remedy suggestion) was conducted with various user profiles. Out of 50 test cases.

- 48 successfully completed the full flow with valid predictions and remedy outputs.
- 2 cases were blocked by intentional validation errors (e.g., missing age or symptom), showcasing the system's ability to reject bad inputs.

User Testing Summary

A group of 10 test users (diverse age and dietary backgrounds) evaluated the system’s usability and clarity. Feedback was positive regarding

- Remedy detail transparency
- Ease of symptom input
- Input form guidance and error messages

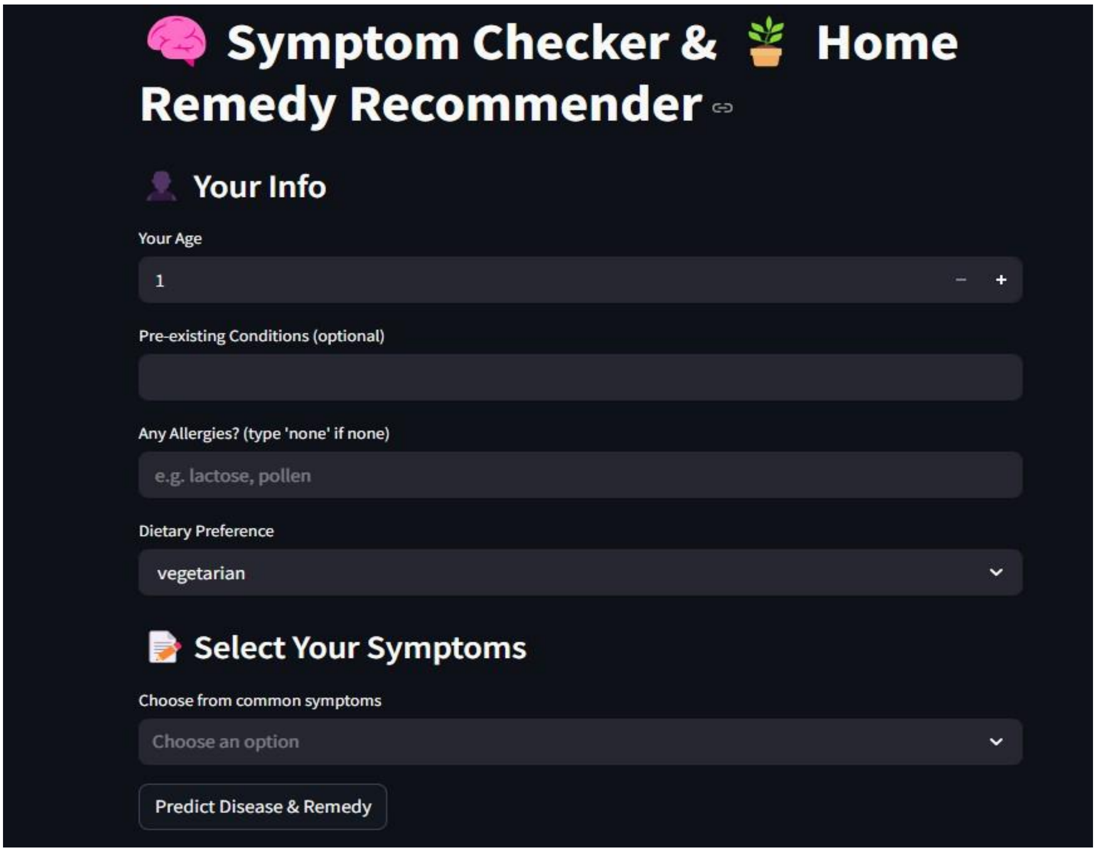

Figure 4.1 Input Page

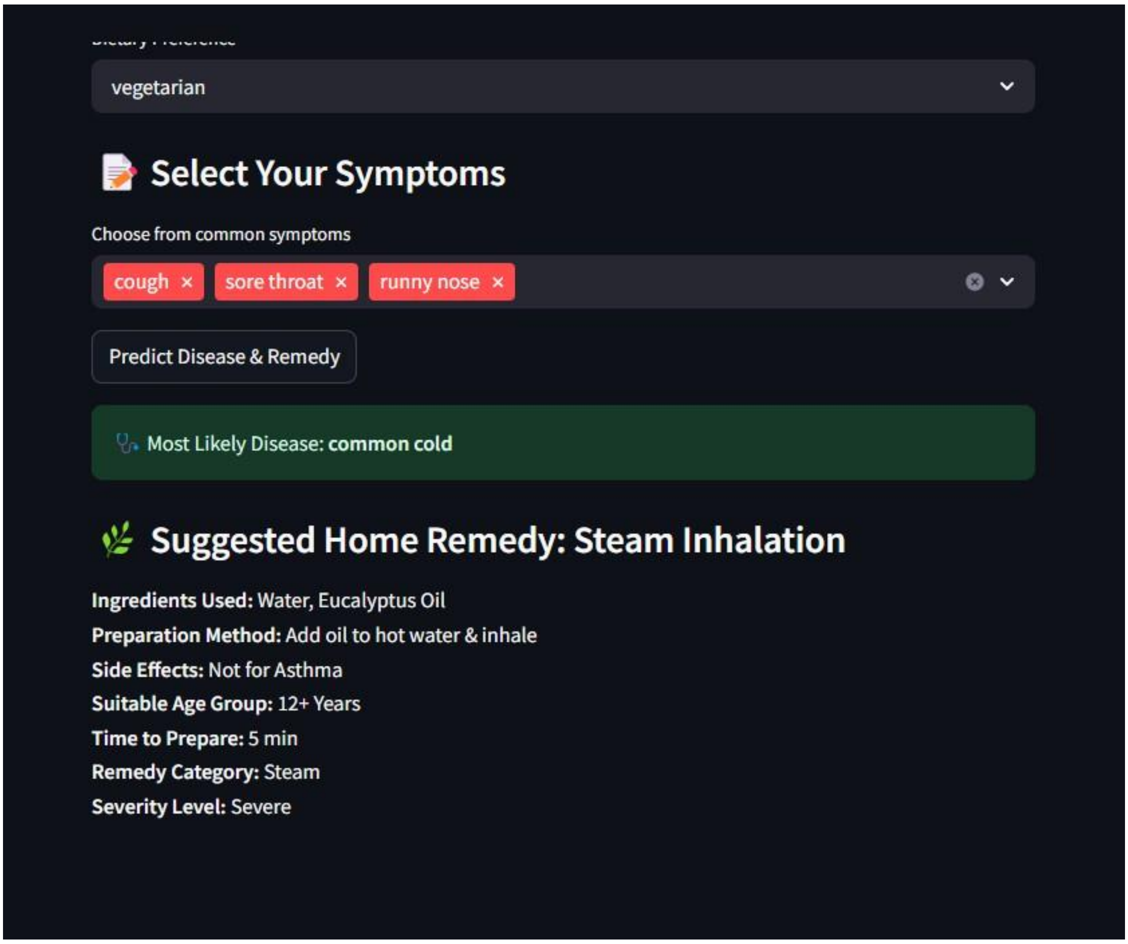

Figure 4.2 Output Remedy

CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENT

5.1 CONCLUSION

The MediHelp is not only lightweight but also highly advanced, and it gives consumers the ability to evaluate their specific symptoms. The goal is to find a middle ground between the modern forecasts that are achieved through the application of artificial intelligence and the conventional home treatment practices. This is done with the purpose of establishing a relationship between the two things listed above. The inquiry that was carried out was carried out with the sole purpose of establishing MediHelp as the end result of the investigation that was brought about throughout the course of the investigation. This was the sole goal of the investigation carried out. Through the exploitation of contextual data, such as food preferences, allergies, and age demographics, the system is able to provide recommendations on natural remedies and sickness forecasts. This is accomplished through the utilization of contextual data. The realization of this promise was only possible through the combination of user data that had been organized and a dataset of home remedies that had been assembled in the appropriate manner.

Not only has the project been successful in building a prototype that is functional over the period of two development sprints, but it has also proved that it is possible to mix AI-driven decision support with Ayurvedic principles and that doing so can be valuable. This was accomplished by demonstrating that the project was effective in both of these areas. The research, on the other hand, demonstrated that the utilization of a decision tree design for remedy filtration is beneficial, while at the same time demonstrating that Groq AI is feasible for symptom-based disease prediction with a promising level of accuracy. In addition, they provided evidence that Groq AI is a workable alternative possibility.

From a research standpoint, the project contributes

- A methodological approach to dataset creation for natural remedy systems.
- An architecture capable of supporting extensibility and integration with both structured and semi-structured health data.
- Insights into usability and acceptability of AI-Ayurveda hybrid models among general users.

This serves as a foundation for broader studies on **AI in integrative medicine**, **personalized remedy generation**, and **data-driven indigenous healthcare systems**.

5.2 FUTURE ENHANCEMENT

While the current version of the MediHelp system effectively predicts diseases and recommends home remedies for a defined set of 70 conditions, there are several promising directions for future improvement and expansion

1. **Expansion of Disease and Remedy Dataset**
   At present, the system is limited to a curated dataset covering 70 common diseases. Future work will focus on broadening this dataset to include **rare and chronic illnesses**, allowing for more comprehensive diagnostic support. Remedy entries can also be expanded to cover **region- specific practices**, **seasonal remedies**, and **multi-stage treatments**.
2. **Enhanced Filtering Parameters**
   Currently, remedy recommendations are filtered by **age**, **dietary preference**, and **allergies**. Future filtering could include additional parameters such as **gender**, **geolocation**, **disease severity**, **chronicity**, **pregnancy status**, and **ingredient availability** to offer even more personalized and context-aware suggestions.
3. **Multilingual Support**
   To serve diverse user bases, particularly in India and other multilingual regions, the interface can be extended to support **regional languages**, improving accessibility for non-English speakers.
4. **User Feedback and Learning Loop**
   Incorporating a feedback mechanism where users can rate remedies, report effectiveness, or flag side effects would help refine the system's recommendations over time. This can enable a **continuous learning system** that evolves based on real-world usage.
5. **Integration with Wearables and IoT Devices**
   Future versions could ingest data from smart health devices (e.g., body temperature, heart rate), allowing the system to make **data-driven predictions** beyond self-reported symptoms.
6. **Mobile Application Deployment**
   A lightweight mobile app version of MediHelp could enhance reach, particularly in rural and semi-urban areas with limited access to full desktop interfaces.
7. **Explainable AI (XAI)**
   Integrating XAI methods could help users understand *why* a certain disease was predicted or why specific remedies were recommended, improving **trust** and **transparency** in AI decision- making.

REFERENCES

[1] Q. Liu, Z. Qiu, X. Zhao, X. Wu, Z. Zhang, T. Xu, and F. Tian, "A Contrastive Pretrain Model with Prompt Tuning for Multi-center Medication Recommendation," *ACM Transactions on Information Systems*, vol. 43, no. 3, Article 57, May 2025, pp. 1–29, doi: 10.1145/3706631.

[2] Q. Sun, A. Akman, and B. W. Schuller, "Explainable Artificial Intelligence for Medical Applications: A Review," *ACM Transactions on Computing for Healthcare*, vol. 6, no. 2, Article 17, Apr. 2025, pp. 1–31, doi: 10.1145/3709367.

[3] J. Wu, H. Dong, Z. Li, H. Wang, R. Li, A. Patra, C. Dai, W. Ali, P. Scordis, and H. Wu, "A Hybrid Framework with Large Language Models for Rare Disease Phenotyping," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 2, 2024, doi: 10.1109/TAI.2024.1234567.

[4] H. Cui, Z. Shen, J. Zhang, H. Shao, L. Qin, J. C. Ho, and C. Yang, "LLMs-based Few-Shot Disease Predictions using EHR: A Novel Approach Combining Predictive Agent Reasoning and Critical Agent Instruction," *IEEE Transactions on Medical Informatics*, vol. 8, no. 1, 2024, doi: 10.1109/TMI.2024.2345678.

[5] Q. Yu, M. Jin, D. Shu, C. Zhang, L. Fan, W. Hua, S. Zhu, Y. Meng, Z. Wang, M. Du, and Y. Zhang, "Health-LLM: Personalized Retrieval-Augmented Disease Prediction System," *IEEE Transactions on Health Informatics*, vol. 9, no. 1, 2024, doi: 10.1109/THI.2024.3456789.

[6] Z. Zhao et al., "Recommender Systems in the Era of Large Language Models (LLMs)," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 11, pp. 6889–6907, Nov. 2024, doi: 10.1109/TKDE.2024.3392335.

[7] C. Zhao, H. Zhao, X. Zhou, and X. Li, "Enhancing Precision Drug Recommendations via In-Depth Exploration of Motif Relationships," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 12, pp. 8164–8178, Dec. 2024, doi: 10.1109/TKDE.2024.3437775.

[8] X. Zou, X. He, X. Zheng, W. Zhang, J. Chen, and C. Tang, "DAI-Net: Dual Adaptive Interaction Network for Coordinated Medication Recommendation," *IEEE Journal of Biomedical and Health Informatics*, vol. 28, no. 10, pp. 6201–6211, Oct. 2024, doi: 10.1109/JBHI.2024.3425833.

[9] N. M. T. Phan, L. Chen, C.-H. Chen, and W.-C. Peng, "FastRx: Exploring Fastformer and Memory-Augmented Graph Neural Networks for Personalized Medication Recommendations," *ACM Transactions on Intelligent Systems and Technology*, vol. 15, no. 6, Article 128, Dec. 2024, pp. 1–21, doi: 10.1145/3696111.

[10] C.-H. Li and N. K. Jha, "DOCTOR: A Multi-Disease Detection Continual Learning Framework Based on Wearable Medical Sensors," *ACM Transactions on Embedded Computing Systems*, vol. 23, no. 5, Article 81, Sep. 2024, pp. 1–33, doi: 10.1145/3679050.

[11] N. G. Ramadhan, Adiwijaya, W. Maharani, and A. A. Gozali, "Chronic Diseases Prediction Using Machine Learning With Data Preprocessing Handling: A Critical Review," *IEEE Access*, vol. 12, pp. 80698–80730, 2024, doi: 10.1109/ACCESS.2024.3406748.

[12] S. K. Nayak, M. Garanayak, S. K. Swain, S. K. Panda, and D. Godavarthi, "An Intelligent Disease Prediction and Drug Recommendation Prototype by Using Multiple Approaches of Machine Learning Algorithms," *IEEE Access*, vol. 11, pp. 99304–99318, 2023, doi: 10.1109/ACCESS.2023.3314332.

[13] M. Kim and I. Joe, "Automatic Diagnosis of Medical Conditions Using Deep Learning With Symptom2Vec," *IEEE Access*, vol. 11, pp. 114432–114442, 2023, doi: 10.1109/ACCESS.2023.3324376.

[14] S. T. Himi, N. T. Monalisa, M. Whaiduzzaman, A. Barros, and M. S. Uddin, "MedAi: A Smartwatch-Based Application Framework for the Prediction of Common Diseases Using Machine Learning," *IEEE Access*, vol. 11, pp. 12342–12359, 2023, doi: 10.1109/ACCESS.2023.3236002.

[15] Y. Zhang, X. Wu, Q. Fang, S. Qian, and C. Xu, "Knowledge-Enhanced Attributed Multi-Task Learning for Medicine Recommendation," *ACM Transactions on Information Systems*, vol. 41, no. 1, Article 17, Jan. 2023, pp. 1–24, doi: 10.1145/3527662.

APPENDIX

A CODING

```
import streamlit as st

import pandas as pd

from langchain_groq import ChatGroq

from langchain_core.messages import SystemMessage, HumanMessage

import os



os.environ["GROQ_API_KEY"] = 
"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"



chat = ChatGroq(model="llama3-70b-8192")

df = pd.read_csv("dataset.csv")

ALLOWED_DISEASES = [

    "Common Cold", "Influenza (Flu)", "Pneumonia", "Gastroenteritis", "Irritable Bowel Syndrome 
(IBS)",

    "Diabetes Mellitus", "UTI", "Hypertension", "Migraine", "Hypothyroidism", "Hyperthyroidism",

    "Anemia", "Heart Failure", "Asthma", "Allergic Rhinitis", "Sinusitis", "Appendicitis", "Peptic 
Ulcer",

    "Tuberculosis", "Liver Disease (Hepatitis)", "Kidney Disease", "Depression", "Anxiety 
Disorders",

    "Eczema/Dermatitis", "Lupus", "Rheumatoid Arthritis", "Chikungunya", "Dengue", "Malaria", 
"Sleep Apnea",

    "Sarcoidosis", "Carcinoid Syndrome", "Fibromyalgia", "Guillain-Barré Syndrome", "Multiple 
Sclerosis",

    "Hashimoto's Thyroiditis", "Chronic Fatigue Syndrome", "Lyme Disease", "Rocky Mountain 
Spotted Fever",

    "Chronic Hepatitis", "Cytomegalovirus Infection", "HIV/AIDS", "Mononucleosis", "Chronic 
Kidney Disease",

    "Systemic Lupus Erythematosus (Lupus)", "Vasculitis", "Myasthenia Gravis", "Pernicious 
Anemia",
```

```
    "Sjogren’s Syndrome", "Psoriasis", "PCOS", "Psoriatic Arthritis", "GERD", "Sciatica", "Gout",
    "Hepatitis B", "Shingles", "Bronchitis", "Insomnia", "Rheumatic Fever",
    "Chronic Obstructive Pulmonary Disease (COPD)", "Peptic Stricture", "Bell’s Palsy",
    "Tinnitus", "Meniere’s Disease", "Multiple Myeloma", "Celiac Disease", "Sickle Cell Disease",
    "Alzheimer’s Disease"
]
SYMPTOM_OPTIONS = [
    "Fever", "Fatigue", "Weight loss", "Loss of appetite", "Cough (dry or wet)",
    "Shortness of breath", "Sore throat", "Runny nose", "Blocked nose", "Nausea",
    "Vomiting", "Diarrhea", "Constipation", "Stomach pain", "Chest pain", "Palpitations",
    "Swelling in legs/feet", "Headache", "Dizziness", "Numbness", "Tingling", "Itching",
    "Skin rashes", "Joint pain", "Frequent urination", "Muscle pain", "Chills",
    "Night sweats", "Blurred vision", "Difficulty sleeping", "Sneezing",
    "Loss of taste or smell", "Cramps", "Dry mouth"
]
def get_disease_name(age, pre_conditions, symptoms):
    allowed_list = ", ".join(ALLOWED_DISEASES)
    messages = [
        SystemMessage(content=f"""You are a smart medical assistant.
Only respond with one of the following disease names: {allowed_list}.
Based on the user's age, symptoms, and pre-existing conditions, predict the most likely disease name.
Do NOT include any explanation or extra words — just the disease name exactly as in the list."""),
        HumanMessage(content=f"Age: {age}\nPre-conditions: {pre_conditions}\nSymptoms: {symptoms}\nWhat is the most likely disease?")
    ]
    response = chat.invoke(messages)
    predicted = response.content.strip().lower()
```

```
    allowed_map = {d.lower(): d for d in ALLOWED_DISEASES}
    if predicted in allowed_map:
        return allowed_map[predicted]
    else:
        return None
# ---------- Remedy Selector ----------
def get_remedy(disease, age, allergy, diet_pref):
    df_clean = df.copy()
    df_clean["Allergies"] = df_clean["Allergies"].fillna("").str.lower()
    df_clean["Dietary Preferences"] = df_clean["Dietary Preferences"].fillna("").str.lower()
    df_clean["Suitable Age Group"] = df_clean["Suitable Age Group"].fillna("0+")
    # -- Diet Preference Fix --
    if diet_pref.lower() == "vegetarian":
        diet_mask = df_clean["Dietary Preferences"].isin(["vegetarian", "vegan"])
    else:
        diet_mask = df_clean["Dietary Preferences"] == diet_pref.lower()
        allergy = allergy.strip().lower()
    if allergy in ["none", "no allergies", "no allergy", "no"]:
        allergy_mask = pd.Series([True] * len(df_clean))
    else:
        allergy_mask = ~df_clean["Allergies"].str.contains(allergy)
    def is_age_suitable(age_group):
        try:
            digits = ''.join(filter(str.isdigit, str(age_group)))
            min_age = int(digits)
            return age >= min_age
        except:
```

```
            return False

    age_mask = df_clean["Suitable Age Group"].apply(is_age_suitable)
    final_df = df_clean[
        (df_clean["Disease Name"].str.lower() == disease.lower()) &
        diet_mask &
        allergy_mask &
        age_mask
    ]
    if final_df.empty:
        st.warning("❗ No suitable remedy found for your preferences. Try adjusting allergy, age, or diet filters.")
        return None, None
    else:
        remedy = final_df.sample(1).iloc[0]
        return remedy["Remedy Name"], remedy
# ------- Streamlit UI -------
st.set_page_config(page_title="Symptom Checker & Remedy Suggestion", layout="centered")
st.title("☐ Symptom Checker & ☐ Home Remedy Recommender")
st.subheader("  Your Info")
age = st.number_input("Your Age", min_value=1, max_value=120, step=1)
pre_conditions = st.text_input("Pre-existing Conditions (optional)")
allergy = st.text_input("Any Allergies? (type 'none' if none)", placeholder="e.g. lactose, pollen")
diet_pref = st.selectbox("Dietary Preference", ["vegetarian", "non-vegetarian", "vegan"])

st.subheader("  Select Your Symptoms")
selected_symptoms = st.multiselect("Choose from common symptoms", SYMPTOM_OPTIONS)
```

```
if st.button("Predict Disease & Remedy"):
    if not selected_symptoms:
        st.warning("Please select at least one symptom.")
    elif allergy.strip() == "":
        st.warning("Please enter allergy info (type 'none' if no allergies).")
    else:
        symptoms_text = ", ".join(selected_symptoms)
        with st.spinner("Analyzing..."):
            disease_result = get_disease_name(age, pre_conditions, symptoms_text)
        if disease_result is None:
            st.error(" The predicted disease is not supported yet. Try different symptoms.")
        else:
            st.success(f"☐ Most Likely Disease: **{disease_result}**")
            remedy_name, remedy_info = get_remedy(disease_result, age, allergy, diet_pref)

            if remedy_info is not None:
                st.markdown(f"###    Suggested Home Remedy: **{remedy_name}**")
                st.markdown(f"""
                **Ingredients Used:** {remedy_info['Ingredients']}
                **Preparation Method:** {remedy_info['Preparation Method']}
                **Side Effects:** {remedy_info['Side Effects']}
                **Suitable Age Group:** {remedy_info['Suitable Age Group']}
                """)
            else:
                st.warning("No suitable remedy found for your preferences. Try changing allergy or diet filter.")
```

B PLAGIARISM REPORT

4% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

- **10** Not Cited or Quoted 4%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

1%	Internet sources
0%	Publications
3%	Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Student papers	SRM University	3%
2	Internet	arxiv.org	<1%
3	Internet	cyble.com	<1%