PROJECT: 2 REPORT

SmartCare: AI Powered Health Care Recommendation System

SUBMITTED BY

Shreya Swapnil Gade

INFOTACT SOLUTIONS

Domain: Data Science and Machine Learning

Position: Associate L1 Intern

Duration: 3 Months

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to all those who have supported and guided me in completing this project titled "**SmartCare AI Powered HealthCare Recommendation"**: **"Disease and Treatment Prediction using Flask and Machine Learning"**.

First and foremost, I thank **Chandan Mishra Sir**, for providing invaluable suggestions, continuous guidance, and timely feedback throughout the development process.

I also acknowledge the assistance of open-source tools and the Python community, without which implementing this project would not have been feasible.

Shreya Swapnil Gade

Contents

Abstract

The increasing demand for intelligent healthcare solutions has paved the way for the integration of machine learning into web applications for early disease detection and personalized treatment recommendations. This project presents a **Flask-based web application** that predicts diseases based on patient-reported symptoms and suggests appropriate treatments including medicines, physical exercises, and estimated recovery time.

The application consists of two core functionalities:

1. **Disease Prediction**: Based on six user-selected symptoms, a trained machine learning classification model identifies the most probable disease using encoded input features and a pre-trained Random Forest model.

2. **Treatment Recommendation**: Once a disease is identified, the user inputs personal health metrics such as age, heart rate, blood pressure, oxygen level, and past history. A multi-output classifier predicts the most suitable medicines and exercises, while a regression model estimates the expected recovery period.

The backend is powered by Python and Flask, and the models are trained using scikit-learn pipelines that combine text processing (via TF-IDF vectorization) with numerical feature scaling. A secure user authentication system is integrated for managing sessions and access.

This project serves as a demonstration of how data-driven models can support primary healthcare decisions by offering reliable predictions and tailored treatment guidance through a user-friendly web interface.

Chapter- 1 Introduction

1.1 Motivation

In recent years, healthcare has seen a rapid transformation through the application of technology, especially in the fields of artificial intelligence and machine learning. With rising patient loads and limited access to medical professionals, there is a growing need for intelligent systems that can assist in preliminary diagnosis and treatment suggestions.

Many people often search online or self-diagnose their health conditions, which can lead to misinformation or panic. Therefore, an AI-powered system that predicts diseases accurately based on symptoms and recommends scientifically grounded treatments can play a vital role in supporting early intervention and guiding users toward appropriate healthcare decisions.

This project was motivated by the desire to bridge the gap between individuals and accessible healthcare by building a web-based platform that utilizes machine learning models trained on medical data. The application aims to offer users a secure, easy-to-use interface where they can enter symptoms and vital signs to receive disease predictions along with treatment guidance.

By integrating classification and regression models into a Flask-based system, the project demonstrates how modern web technologies and AI can work hand-in-hand to deliver meaningful healthcare insights—even in remote or underserved areas.

1.2 Problem Statement:

In the current healthcare landscape, early detection of diseases and personalized treatment planning remain major challenges—especially in areas with limited access to medical professionals. Patients often lack the medical knowledge required to interpret their symptoms accurately, and online resources may provide misleading or generalized information, which can lead to misdiagnosis or delayed treatment.

Additionally, treatment decisions are typically based on a doctor's evaluation, which may not be readily available to everyone, particularly in rural or under-resourced regions. There is a lack of intelligent systems that combine symptom analysis, patient vitals, and history to provide a holistic suggestion for both diagnosis and treatment.

Thus, the core problem addressed in this project is:

"How can we develop an intelligent, accessible, and reliable system that predicts diseases based on patient symptoms and suggests appropriate treatment plans—including

medicines, exercises, and estimated recovery time—using machine learning and a web interface?"

The proposed system aims to tackle this problem by building:

- A symptom-based disease prediction engine.
- A treatment recommender that utilizes vitals and medical history.
- A secure, user-friendly web application accessible to any user.

This addresses both the need for **accurate preliminary diagnosis** and **personalized treatment suggestions**, helping bridge the gap between patients and early-stage medical support.

Chapter- 2 Literature Review

2.1 Introduction

In recent years, the intersection of healthcare and artificial intelligence has garnered significant attention, leading to the development of decision support systems that assist in disease diagnosis and treatment recommendation. Several research studies and real-world applications have explored the use of machine learning models, particularly in automating diagnosis based on patient symptoms, medical history, and clinical data.

The purpose of this literature review is to explore existing methodologies, frameworks, and tools related to disease prediction and treatment suggestion. It focuses on how various machine learning models—such as Random Forest, Support Vector Machines (SVM), and Neural Networks—have been applied to healthcare datasets to provide diagnostic support. Moreover, it reviews how natural language processing (NLP) and multi-output learning have been used to recommend treatment plans including medication, physical exercises, and recovery estimations.

This chapter highlights the current trends, gaps, and technological advancements in the field of intelligent healthcare systems. It sets the foundation for justifying the need for an integrated web-based solution like the one developed in this project, which combines disease prediction and treatment recommendation through a user-friendly interface.

2.2 Existing Methodologies

Several methodologies have been proposed and implemented in the domain of disease prediction and treatment recommendation using machine learning and artificial intelligence. These methods primarily focus on analyzing medical symptoms, patient history, and physiological parameters to assist in diagnosis and decision-making. Below are some of the widely used approaches:

1. Symptom-Based Disease Classification

Machine learning models such as **Decision Trees**, **Random Forest**, **Naive Bayes**, and **Support Vector Machines (SVM)** have been extensively used to classify diseases based on symptoms. These models learn patterns from labeled datasets containing symptoms and their corresponding diseases to make future predictions.

- **Example**: A Random Forest model trained on symptom vectors to classify common diseases like diabetes, flu, or pneumonia.

2. Rule-Based Expert Systems

These systems use a predefined set of if-else rules derived from medical knowledge bases. While easy to interpret, they lack adaptability and fail when encountering new patterns or ambiguous inputs.

- **Limitation**: Inflexible and do not scale well with large symptom variations.

3. Natural Language Processing (NLP) for Medical Text

Text-based data such as patient history, doctor notes, or medical reports are processed using NLP techniques like **TF-IDF**, **word embeddings (Word2Vec, BERT)**, and **topic modeling**. These features are used for both diagnosis and treatment prediction.

- **Use Case**: Extracting relevant features from unstructured patient history to recommend personalized treatments.

4. Multi-Output Classification for Treatment Suggestion

Treatment prediction often involves suggesting multiple outputs (e.g., medicine, exercises), which cannot be modeled by traditional classifiers. Multi-output models like **MultiOutputClassifier** are used to handle multiple labels at once.

- **Example**: Predicting multiple medicines and exercises simultaneously for a diagnosed disease.

5. Regression Models for Recovery Time Estimation

Recovery duration is a continuous outcome and is best predicted using regression models such as **Linear Regression**, **Random Forest Regressor**, or **Gradient Boosting Regressor**. These models consider patient vitals, age, and disease severity.

6. Web-Based Implementation Using Flask/Django

Several research projects have used **Flask** or **Django** frameworks to build full-stack health applications that allow users to input symptoms and receive predictions in real-time.

- **Advantage**: User-friendly, platform-independent, and can integrate with ML models through APIs.

Chapter- 3 Software Requirement Specification

3.1 Hardware Requirements

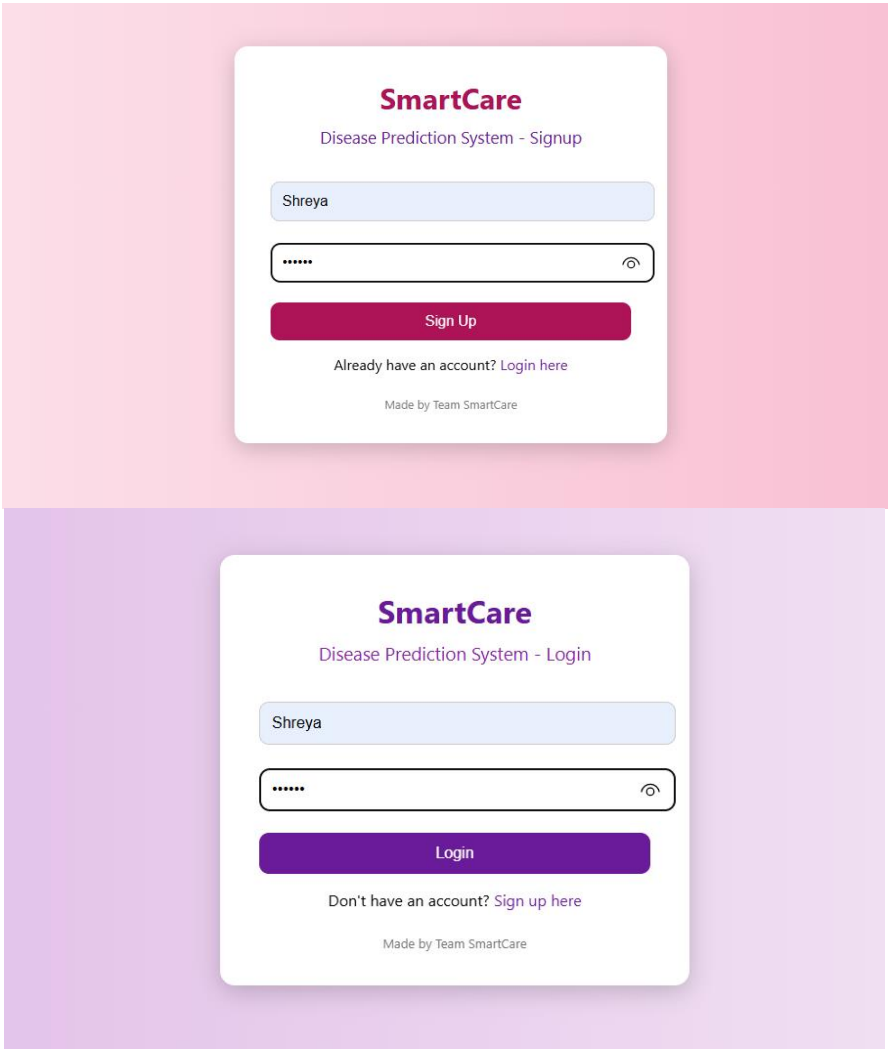
- A computer with an Intel i3 processor or better.
- At least 4 GB of RAM (8 GB is better).
- Around 500 MB of free disk space.
- A screen of 13 inches or more.
- Internet connection to run the web app.
- GPU is optional (only needed for model training).

3.2 Software Requirements

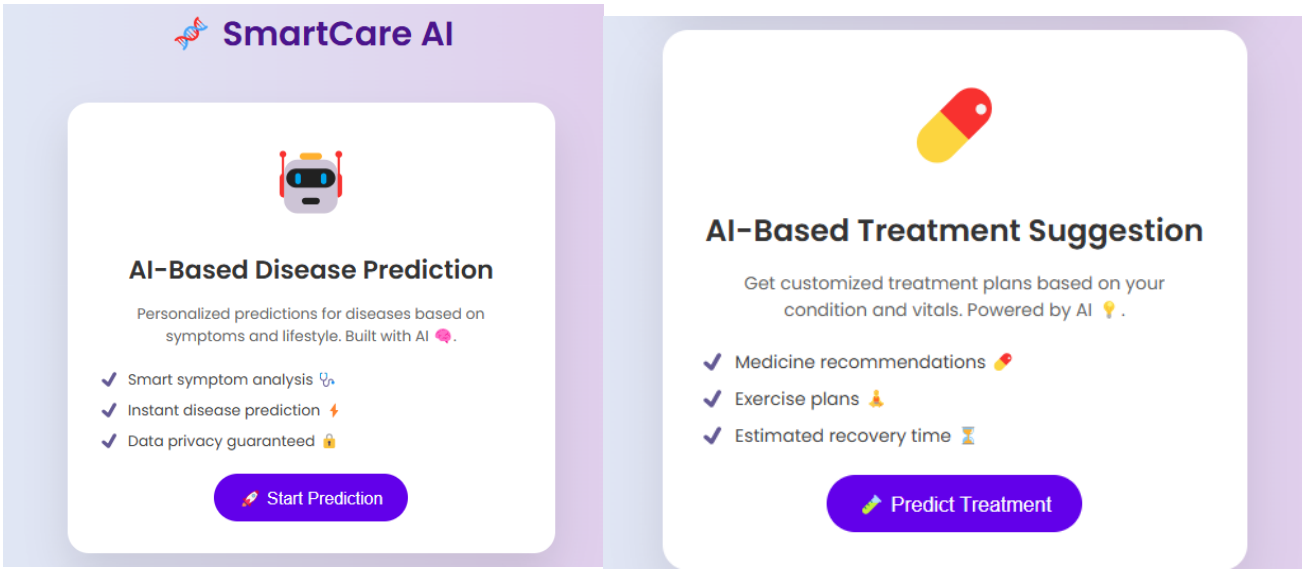
- Windows, Linux, or macOS can be used.
- Python 3.8 or higher is required.
- Flask framework is used for the backend.
- Libraries like scikit-learn, pandas, numpy are used.
- pickle is used to save and load ML models.
- werkzeug is used to hash passwords.
- HTML, CSS, and JavaScript are used for the frontend.
- Google Chrome or any modern browser is recommended.
- No external database is used (data is stored in memory).
- VS Code or any Python IDE can be used to build the project.

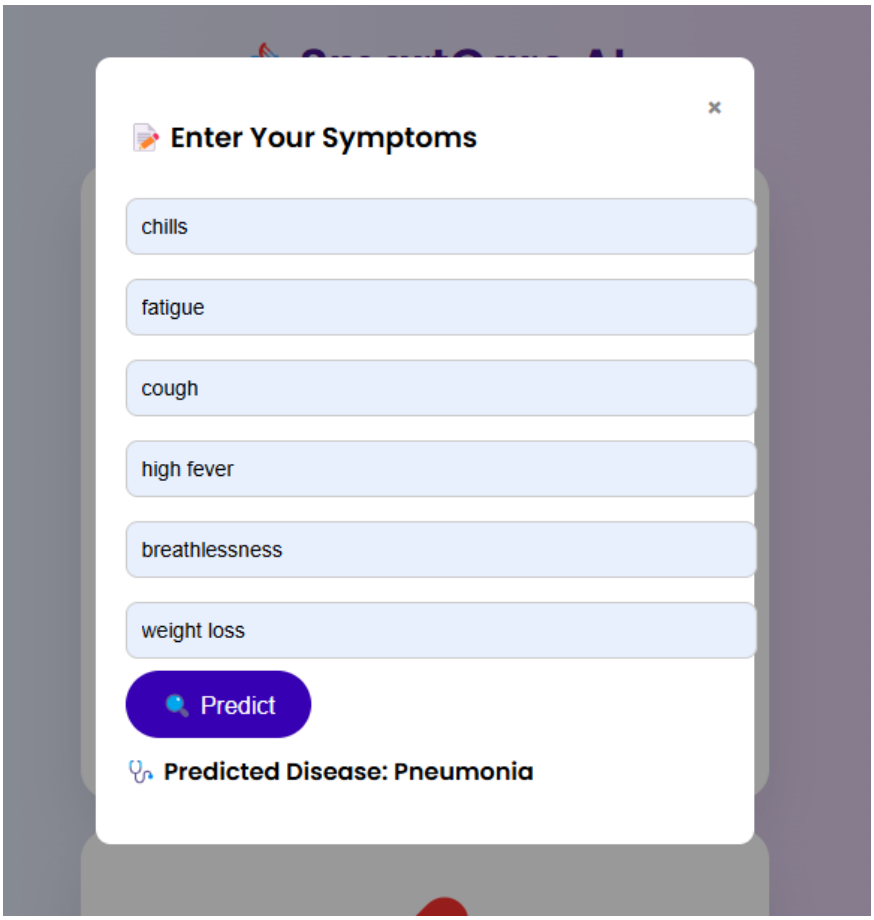
Chapter -5 Forms/jFrames(GUI)

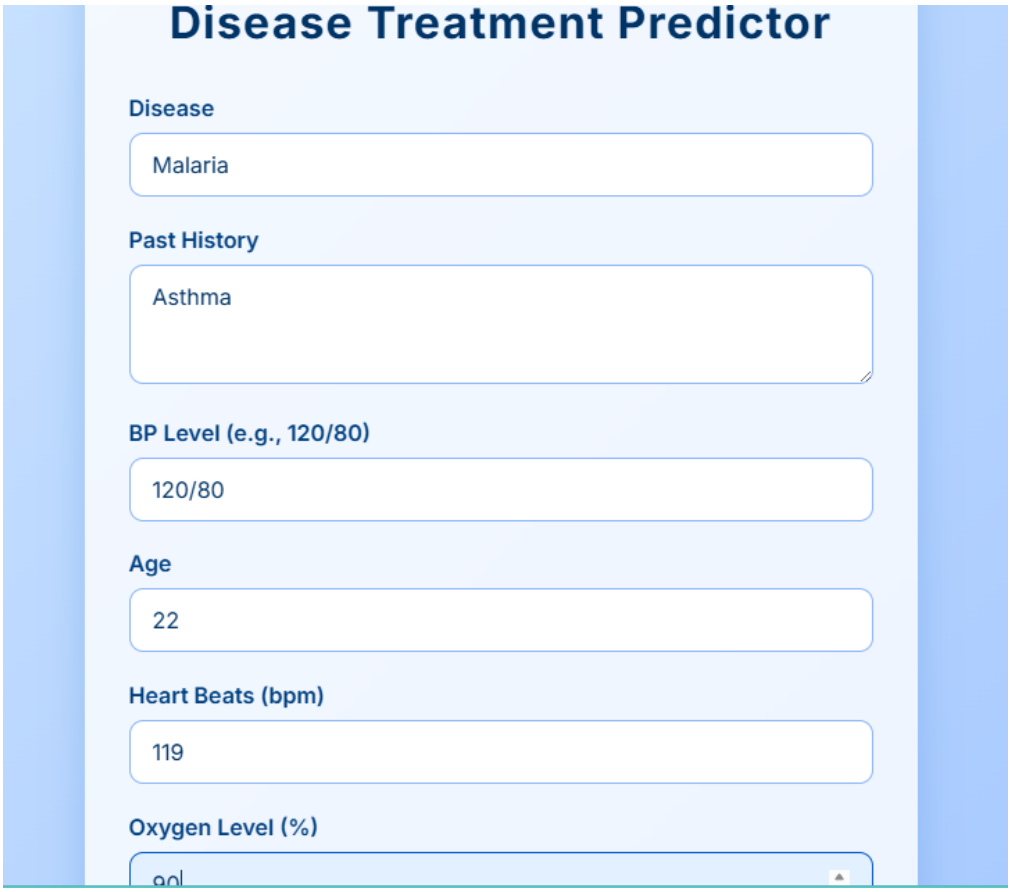
User Authentication

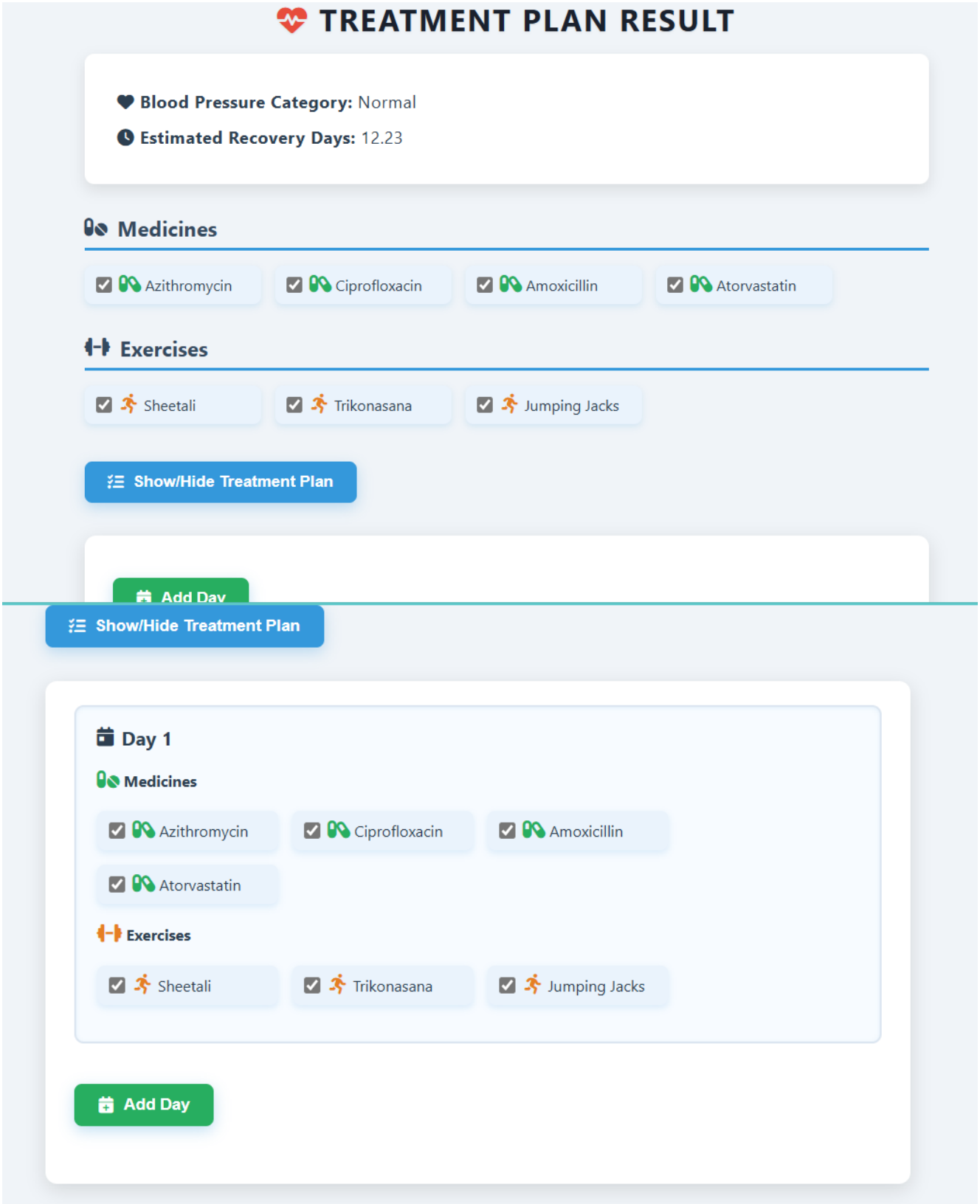


Dashboard




Enter Your Symptoms
chills
fatigue
cough
high fever
breathlessness
weight loss
Predict
Predicted Disease: Pneumonia


Disease Treatment Predictor
Disease
Malaria
Past History
Asthma
BP Level (e.g., 120/80)
120/80
Age
22
Heart Beats (bpm)
119
Oxygen Level (%)


TREATMENT PLAN RESULT
Blood Pressure Category: Normal
Estimated Recovery Days: 12.23
Medicines
Azithromycin
Ciprofloxacin
Amoxicillin
Atorvastatin
Exercises
Sheetali
Trikonasana
Jumping Jacks
Show/Hide Treatment Plan
Add Day
Show/Hide Treatment Plan
Day 1
Medicines
Azithromycin
Ciprofloxacin
Amoxicillin
Atorvastatin
Exercises
Sheetali
Trikonasana
Jumping Jacks
Add Day

Chapter -5 Conclusion

This project successfully demonstrates the integration of machine learning with web technologies to assist in healthcare prediction and planning. The system predicts diseases based on symptoms and suggests treatments such as medicines, exercises, and recovery time based on user vitals and history.

The project was implemented using Python, Flask, and machine learning models trained on a medical dataset. It includes secure user authentication and a simple, interactive web interface. Two ML pipelines were developed—one for classification (treatment) and one for regression (recovery days), which work together to provide personalized results.

This system can be especially helpful in areas with limited medical access, offering quick and intelligent support for early diagnosis and treatment recommendations. It can be further improved by integrating a real database, expanding the dataset, and including doctor suggestions or telemedicine options.

Overall, this project showcases how AI can support and enhance the healthcare process through automation, accessibility, and intelligence.

Chapter- 5 References

1. Scikit-learn Documentation – Machine Learning in Python https://scikit-learn.org/
2. Flask Documentation – Web Development, Python https://flask.palletsprojects.com/
3. Pandas Documentation – Data Analysis Library https://pandas.pydata.org/
4. NumPy Documentation – Scientific Computing in Python https://numpy.org/
5. TF-IDF Vectorization – Wikipedia https://en.wikipedia.org/wiki/Tf%E2%80%93idf
6. MultiOutputClassifier – scikit-learn API Reference https://scikit-learn.org/stable/modules/generated/sklearn.multioutput.MultiOutputClassifier.html
7. Kaggle – Medical Datasets and Resources https://www.kaggle.com/
8. Python Pickle Module – Data Serialization https://docs.python.org/3/library/pickle.html
9. WHO – Disease and Treatment Guidelines https://www.who.int/
10. Stack Overflow – Developer Community Discussions https://stackoverflow.com/