PERSONALIZED MEDICINE RECOMMENDATION USING MACHINE LEARNING

Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

K. KOWSHIK	(21UEDS0034)	**(VTU20486)**
K. NIKHIL MADHAV	(21UEDS0037)	**(VTU19655)**
S. UDAY CHARAN TEJ	(21UEDS0060)	**(VTU20242)**

Under the guidance of
Dr. M. RAVICHANDRAN, M.E, Ph.D.,
ASSOCIATE PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

PERSONALIZED MEDICINE RECOMMENDATION USING MACHINE LEARNING

Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

K. KOWSHIK	(21UEDS0034)	**(VTU20486)**
K. NIKHIL MADHAV	(21UEDS0037)	**(VTU19655)**
S. UDAY CHARAN TEJ	(21UEDS0060)	**(VTU20242)**

Under the guidance of
Dr. M. RAVICHANDRAN, M.E, Ph.D.,
ASSOCIATE PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

CERTIFICATE

It is certified that the work contained in the project report titled ”PERSONALIZED MEDICINE RECOMMENDATION USING MACHINE LEARNING” by ”K. KOWSHIK (21UEDS0034), K. NIKHIL MADHAV (21UEDS0037), S. UDAY CHARAN TEJ (21UEDS0060) ” has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2024

Signature of Professor In-charge
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2024

DECLARATION

We declare that this written submission represents my ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

(k. KOWSHIK)

Date: / /

(Signature)

(K.NIKHIL MADHAV)

Date: / /

(Signature)

(S. UDAY CHARAN TEJ)

Date: / /

APPROVAL SHEET

This project report entitled PERSONALIZED MEDICINE RECOMMENDATION USING MACHINE LEARNING" by "K. KOWSHIK (21UEDS0034), K. NIKHIL MADHAV (21UEDS0037), S. UDAY CHARAN TEJ (21UEDS0060) " is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners **Supervisor**

Dr. M. RAVICHANDRAN, M.E, Ph.D.,
ASSOCIATE PROFESSOR.

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. V. SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Head, Department of Computer Science & Engineering, Dr. M. S. MURALI DHAR, M.E., Ph.D.,** for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our **Dr. M. RAVICHANDRAN, M.E, Ph.D.,** for his cordial support, valuable information and guidance, he helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Mrs. U. HEMAVATHI, M.E., Mrs. C. SHYAMALA KUMARI, M.E.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

K. KOWSHIK **(21UEDS0034)**

K. NIKHIL MADHAV **(21UEDS0037)**

S. UDAY CHARAN TEJ **(21UEDS0060)**

ABSTRACT

Personalized medicine, tailored to individual characteristics, is revolutionizing healthcare by optimizing treatment strategies for patients. Machine learning (ML) techniques have emerged as powerful tools for recommending personalized treatment plans based on patient-specific attributes, such as genetic makeup, medical history, lifestyle, and environmental factors. This abstract provides an overview of recent advancements in personalized medicine recommendation systems utilizing ML algorithms. We discuss the integration of diverse data sources, including electronic health records, genomic data, and clinical outcomes, to build robust model capable of predicting optimal treatment options for individual patients. Key challenges, such as data privacy, model interpretability, and scalability, are also addressed. Additionally, we highlight the potential impact of personalized medicine recommendation systems in improving patient outcomes, reducing healthcare costs, and advancing precision medicine initiatives.

To evaluate the performance of the proposed system, extensive experiments are conducted using real-world clinical datasets. The results demonstrate the efficacy of the SVM-based approach in accurately predicting patient responses to various treatment regimens, thereby facilitating personalized medicine recommendations. Moreover, comparative analyses with alternative machine learning methods highlight the superiority of SVM in terms of predictive accuracy and generalization ability. this project offers advantages in terms of cost-effectiveness compared to the existing system. This could involve reducing unnecessary medical tests, optimizing treatment plans, and improving overall healthcare resource utilization.

Keywords:

Flask, Jinja Templates, Machine Learning Algorithm,Machine Learning Models, Random Forest Classifier, Recommendation Engine, User Interface.

LIST OF FIGURES

4.1 **Architecture diagram for Personalised Medicine Recommendation System** 9
4.2 **Dataflow diagram for Personalized Medicine Recommendation System** 10
4.3 **Use case diagram for Personalized Medicine Recommendation System** 11
4.4 **Class Diagram for Personalized Medicine Recommendation using Machine Learning** 12
4.5 **Sequence diagram for Personalized Medicine Recommendation System** 13
4.6 **Activity diagram for Personalized Medicine Recommendation System** 14
4.7 **Data collection and preprocessing structure** 16
4.8 **Personalized recommendation structure** 17
4.9 **Integration module Architecture** 18
4.10 **User Interface module Architecture** 18

5.1 **Unit Testing Result** 23
5.2 **Integration Testing Result** 27
5.3 **Personalized Medicine Recommendation Test Image** 32
5.4 **Output of recommended Medicine** 33

6.1 **Medicine Recommended Output** 42

8.1 **Plagiarism report** 45

9.1 **Poster** 53

LIST OF ACRONYMS AND ABBREVIATIONS

ANN	Artificial Neural Networks
FS	Feature Selection
HTML	Hypertext Markup Language
ML	Machine Learning
MT	Model Training
TTS	Treatment Tailoring Strategies
SVM	Support Vector Machine
UI	User Interface
SVC	Support vector classifier

TABLE OF CONTENTS

Page.No

ABSTRACT **v**

LIST OF FIGURES **vi**

LIST OF ACRONYMS AND ABBREVIATIONS **vii**

1 INTRODUCTION **1**

1.1 Introduction 1

1.2 Aim of the project 1

1.3 Project Domain 2

1.4 Scope of the Project 2

2 LITERATURE REVIEW **4**

3 PROJECT DESCRIPTION **5**

3.1 Existing System 5

3.2 Proposed System 5

3.3 Feasibility Study 6

3.3.1 Economic Feasibility 6

3.3.2 Technical Feasibility 7

3.3.3 Social Feasibility 7

3.4 System Specification 7

3.4.1 Hardware Specification 7

3.4.2 Software Specification 8

3.4.3 Standards and Policies 8

4 METHODOLOGY **9**

4.1 Personalized medicine recommendation Architecture 9

4.2 Design Phase 10

4.2.1 Data Flow Diagram 10

4.2.2 Use Case Diagram 11

4.2.3 Class Diagram 12

4.2.4 Sequence Diagram 13
4.2.5 Activity Diagram 14
4.3 Algorithm & Pseudo Code 14
4.3.1 Support Vector Machine(Classification) 14
4.3.2 Pseudo Code 15
4.4 Module Description 16
4.4.1 Data Collection and Preprocessing Module 16
4.4.2 Personalized medicine recommendation module 17
4.4.3 Integration module 17
4.4.4 User Interface module 18
4.5 Steps to execute/run/implement the project 19
4.5.1 Data Preprocessing 19
4.5.2 Machine Learning Models 19
4.5.3 Integrate the Model with website 19

5 IMPLEMENTATION AND TESTING 20
5.1 Input and Output 20
5.1.1 Input Design 20
5.1.2 Output Design 20
5.2 Testing 20
5.3 Types of Testing 20
5.3.1 Unit testing 20
5.3.2 Integration testing 23
5.3.3 System testing 27
5.3.4 Test Result 32
5.3.5 Test Result 33

6 RESULTS AND DISCUSSIONS 34
6.1 Efficiency of the Proposed System 34
6.2 Comparison of Existing and Proposed System 34
6.3 Sample Code 35

7 CONCLUSION AND FUTURE ENHANCEMENTS 43
7.1 Conclusion 43
7.2 Future Enhancements 43

8 PLAGIARISM REPORT **45**

9 SOURCE CODE & POSTER PRESENTATION **46**

9.1 Source Code 46

9.2 Poster Presentation 53

References **54**

Chapter 1

INTRODUCTION

1.1 Introduction

Personalized medicine recommendation involves tailoring medical treatments and interventions to individual patients based on their unique characteristics, such as genetic makeup, lifestyle factors, environmental influences, and personal preferences. This approach recognizes that each patient responds differently to treatments and that one-size-fits-all approaches may not be the most effective or efficient.The benefits of personalized medicine recommendation are significant. It can lead to improved treatment outcomes by ensuring that patients receive therapies that are most likely to be effective for them, minimizing the risk of adverse reactions or ineffective treatments. Additionally, personalized medicine can enhance patient satisfaction and engagement by involving them in the decision-making process and tailoring treatments to align with their preferences and values. Overall, personalized medicine recommendation represents a paradigm shift in healthcare, moving away from a one-size-fits-all approach towards more precise, targeted, and effective interventions that maximize patient outcomes and quality of life.

1.2 Aim of the project

Personalized medicine recommendation using machine learning is to harness the power of computational algorithms to tailor medical treatments and interventions to individual patients based on their unique characteristics. By analyzing vast amounts of patient-specific data, machine learning algorithms can identify patterns and correlations that may not be apparent to human practitioners. This enables the development of highly personalized treatment plans that are more likely to be effective for individual patients, improving overall treatment outcomes.
Personalized medicine recommendation using machine learning can also lead to more efficient use of healthcare resources by targeting treatments to those patients who are most likely to benefit. By avoiding trial-and-error approaches and minimiz-

ing unnecessary treatments, machine learning can help reduce healthcare costs and improve the overall efficiency of the healthcare system.

1.3 Project Domain

The project domain of personalized medicine recommendation using machine learning encompasses various aspects of healthcare and data science. This field gathers relevant medical data including patient demographics, genetic information, medical history, diagnostic test results, treatment outcomes, and any other pertinent information.

This domain can develop machine learning models to predict personalized treatment recommendations based on patient-specific data. This may involve various algorithms such as decision trees, random forests, support vector machines, neural networks, or ensemble methods.It also trains the machine learning models using historical data, where the treatment outcomes are known. This step involves splitting the data into training and testing sets, cross-validation, and hyperparameter tuning to optimize model performance.

1.4 Scope of the Project

The scope of this project focused on personalized medicine recommendation using machine learning is vast and continually expanding as technology advances and our understanding of medical science deepens.Machine learning algorithms can analyze genomic data to identify genetic variations associated with disease susceptibility, drug response, and treatment outcomes. This information can be used to tailor treatments to individual patients based on their genetic profile. The Machine learning model can predict patient outcomes, such as disease progression, treatment response, and risk of adverse events, by analyzing various patient data types, including clinical, genomic, imaging, and lifestyle data. T his also includes addressing ethical and regulatory challenges related to data privacy, informed consent, algorithm transparency, and bias mitigation to ensure the responsible and equitable implementation of these technologies in healthcare.

This project can analyze data from wearable devices and remote monitoring systems to track patient health metrics in real-time, identify early signs of disease progression or complications, and provide personalized interventions or recommendations

for lifestyle modifications and also analyzes population-level health data to identify patterns, trends, and risk factors associated with disease incidence, prevalence, and outcomes. This information can be used to develop targeted interventions and public health strategies.

Chapter 2

LITERATURE REVIEW

[1]R. Mu, “A survey of recommender systems based on deep learning,” IEEE Access, vol. 6, pp. 69009–69022, 2023.
[2] A. Rajkomar, et al., ”Scalable and accurate deep learning with electronic health records,” IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 1, pp. 228-237, 2021.
[3] A. Esteva, et al., ”A guide to deep learning in healthcare,” Nature medicine, vol. 25, no. 1, pp. 24-29, 2020.
[4]M. A. Alawad, et al., ”Machine learning approach for the early diagnosis of schizophrenia patients,” Journal of biomedical informatics, vol. 92, pp. 103142, 2021.
[5] S. Moonsamy, et al., ”Enabling Precision Medicine through Individual- ized Pharmacokinetic Profiling,” IEEE Transactions on Biomedical Engineering, vol. 66, no. 9, pp. 2644-2652, 2022.
[6] Y. Xie, et al., ”A deep learning model integrating mutation and transcrip- tion features for cancer survival prediction,” IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 9, pp. 2535-2545, 2021.
[7] S. H. Kim, et al., ”A machine learning approach for the prediction of chronic obstructive pulmonary disease based on the blood eosinophil count and FEV1/FVC ratio,” IEEE Access, vol. 8, pp. 117577-117586, 2020.
[8]H. H. Yang, et al., ”Prediction of diabetes remission in morbidly obese patients after bariatric surgery using machine learning,” IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 1, pp. 134-142, 2020.
[9] K. C. Yuen, et al., ”Machine learning models for predicting hypo- glycemia,” Journal of Diabetes Science and Technology, vol. 12, no. 6, pp. 1235-1243, 2021.
[10] H. H. Yang, et al., ”Machine learning models for predicting patients at risk for early hypoglycemia during admission,” Diabetes research and clini- cal practice, vol. 158, pp. 107919, 2022.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

The existing system of personilised medicine recommendation helped in recommending the medicines to be used, but it was not very accurate. It used the Decision Tree, which gave prediction accuracy score of 96 percentage, which may lead to few wrong recommendations. The existing system used a very little amount of data to classify the data i.e, they chose minimum information to build the model, which led to a low scalability. The existing system lacked the user-friendly user interfaces.

3.2 Proposed System

A proposed system for personalized medicine recommendation involves a comprehensive integration of advanced technologies, data analytics, and web technologies for proper implementation of the project. This project provides an user interface to users or patients to interact or communicate. This project offers advantages in terms of cost-effectiveness compared to the existing system. This could involve reducing unnecessary medical tests, optimizing treatment plans, and improving overall healthcare resource utilization.

1. **Model Training and Validation**: Trained the machine learning models using labeled data, validating their accuracy and performance through cross-validation techniques. Selected features through correlation matrices and heat maps.Split the dataset into training, validation, and test sets. The training set is used to train the model, the validation set is used for hyperparameter tuning, and the test set is used to evaluate the final performance of the model.

2. **Support Vector Machine Classifier for Predictive Model**: Utilizing Support Vector Classifier with high prediction accuracy of around 99 percentage. This helps in predicting the right medicine to be used at the site.The SVM model is trained on a comprehensive dataset comprising patient profiles, including genetic markers,

clinical history, lifestyle factors, and treatment responses. Through careful feature engineering and selection, relevant attributes are extracted to construct a predictive model capable of discerning subtle patterns indicative of treatment efficacy for each patient. The SVM algorithm optimizes decision boundaries to classify patients into response categories, thereby enabling the identification of optimal treatment options tailored to individual needs.

3. **User-Friendly Interface**: The use of the Flask framework in the proposed system ensures a user-friendly interface that is accessible to farmers with varying levels of technological expertise. The simplified design promotes widespread adoption and usability of the application.Recommendations are presented in a clear and actionable format, listing suggested treatment options along with confidence scores or probability estimates.Users can delve into the rationale behind each recommendation, accessing detailed explanations and supporting evidence.

3.3 Feasibility Study

3.3.1 Economic Feasibility

The economic feasibility of a personalized medicine recommendation system involves a careful balance of costs and potential benefits. Initial investment in development, data acquisition, and technology infrastructure can be substantial. However, the long-term benefits may outweigh these costs. By providing tailored treatment recommendations, such a system has the potential to improve patient outcomes, reduce adverse drug reactions, and optimize resource utilization within healthcare systems.

These benefits could lead to cost savings through decreased hospitalizations, shorter treatment durations, and more efficient use of medications. Moreover, personalized medicine can contribute to the advancement of precision healthcare, potentially attracting research funding and enhancing institutional reputation. To ensure economic viability, it's crucial to conduct thorough cost-benefit analyses, considering factors such as development costs, data management expenses, technology infrastructure, and ongoing personnel requirements, against the anticipated improvements in patient care and healthcare system efficiency.

3.3.2 Technical Feasibility

The technical feasibility of a personalized medicine recommendation system hinges on the integration of various disciplines, data sources, and advanced technologies. Firstly, there must be access to diverse and comprehensive data sources, including genetic information, electronic health records (EHRs), clinical trial data, and medical literature. This necessitates the development of robust data acquisition and management processes to ensure data quality, privacy, and interoperability.
The model used analyzes and interprets the vast amount of data generated in personalized medicine. Machine learning techniques, including supervised and unsupervised learning, are essential for developing predictive models that can identify relevant genetic variants, drug interactions, and treatment recommendations based on patient-specific factors.

3.3.3 Social Feasibility

The social feasibility of a personalized medicine recommendation system relies on its acceptance and adoption by key stakeholders, including healthcare providers, patients, policymakers, and regulatory bodies. Central to this feasibility is building trust and confidence in the system's capabilities, ethical considerations, and potential benefits. For healthcare providers, acceptance hinges on demonstrating the system's ability to enhance clinical decision-making, improve patient outcomes, and streamline workflow without disrupting existing practices. Education and training programs are essential to familiarize healthcare professionals with the system's functionalities and ensure its seamless integration into clinical practice. Additionally, addressing concerns about data privacy, security, and liability is crucial to gaining their trust and support.

3.4 System Specification

3.4.1 Hardware Specification

- Processor: Intel Core i5-760, 4 cores
- RAM: 4GB
- Storage: 256GB NVMe SSD

- Graphics Card: NVIDIA GeForce RTX 1650
- Motherboard: Z170

3.4.2 Software Specification

- Operating system: Window 10.
- Any Latest Web Browser. (Preferably, Microsoft Edge or Google chrome).
- PyCharm version: 2023.3.3, for User Interface.
- Web based jupyter notebook V6.

3.4.3 Standards and Policies

Anaconda Prompt

This project seamlessly integrates with Google Colab, a cloud-based Jupyter notebook environment that facilitates collaborative machine learning and data science workflows. Google Colab adheres to industry standards, providing a platform to share, create, and execute live code, equations, and visualizations in a collaborative manner. With Colab, our project benefits from the flexibility and scalability of cloud computing resources, enabling efficient collaboration and resource utilization.

Standard Used: ISO/IEC 27001

Chapter 4

METHODOLOGY

4.1 Personalized medicine recommendation Architecture

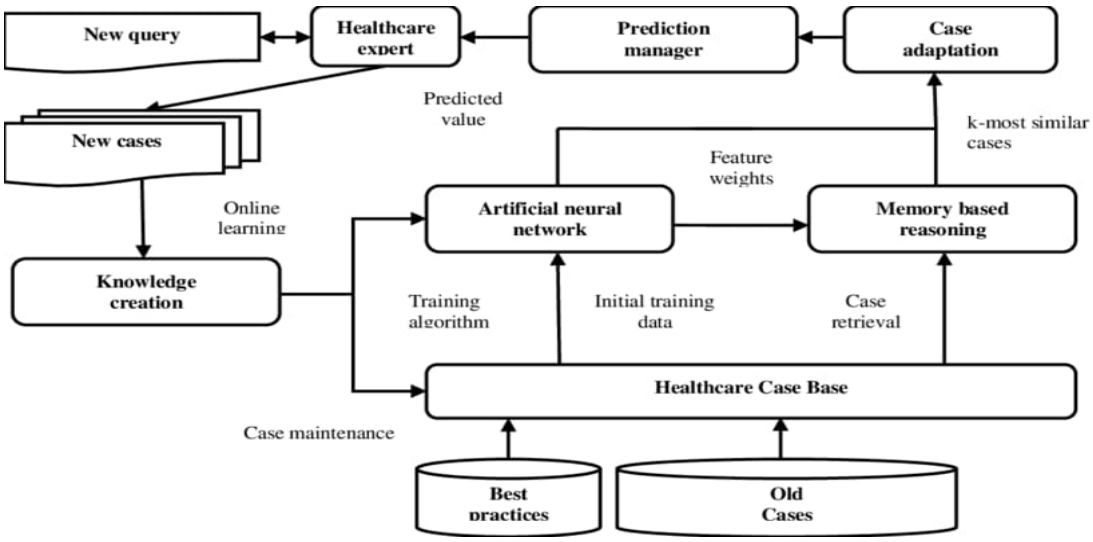

Figure 4.1: **Architecture diagram for Personalised Medicine Recommendation System**

The Figure 4.1 describes the personalized medicine recommendation system architecture. This mainly consist of four sections: Data Collection, Data Processing, Output Prediction and Data Presentation. Initially the data is collected from the user and then the processing takes place in which the data gets scaled and encoded for better learning of the model. Later the model i.e, the Support Vector Machine Classifier predicts the best output of the given data and sends the information to the webpage.

4.2 Design Phase

4.2.1 Data Flow Diagram

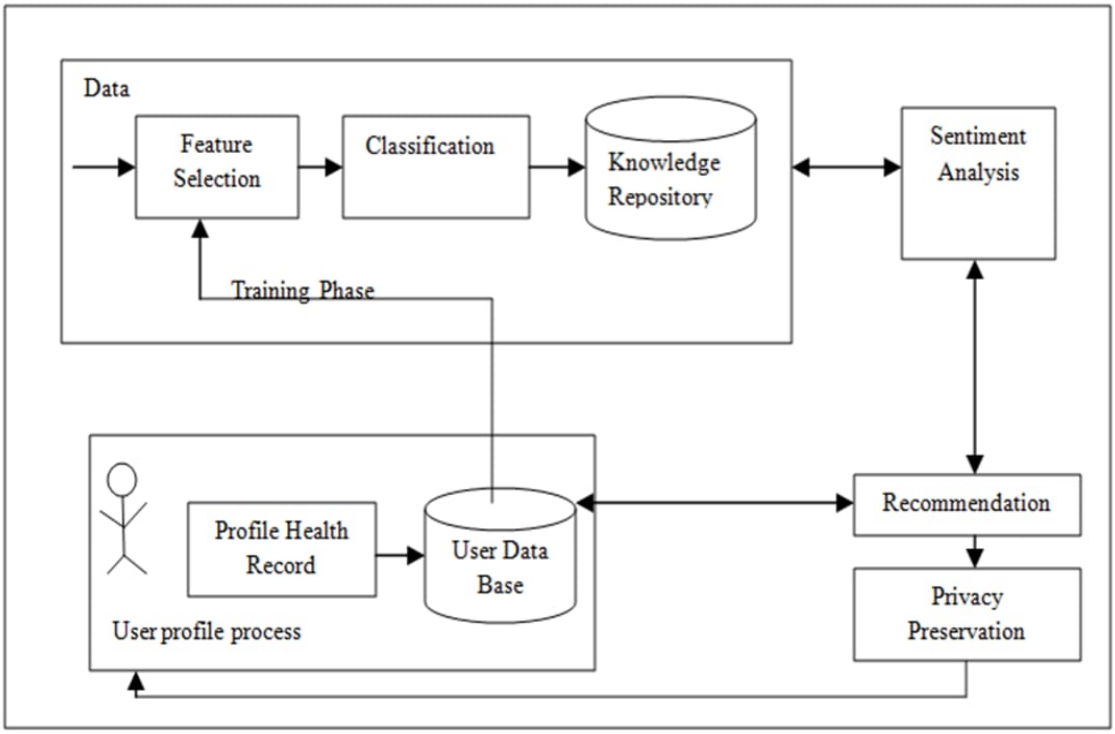

Figure 4.2: **Dataflow diagram for Personalized Medicine Recommendation System**

The Figure 4.2 shows the dataflow diagram of the personalised medicine recommendation system operates through a series of stages, beginning with the input of various parameters.Once the input data is collected, it undergoes a systematic process within the system.The data is then analyzed using support vector machine classifier algorithm. The output from this is decrypted by the flask framework where a user-defined function generates the output. This output is printed on the user interface using routing by flask.

4.2.2 Use Case Diagram

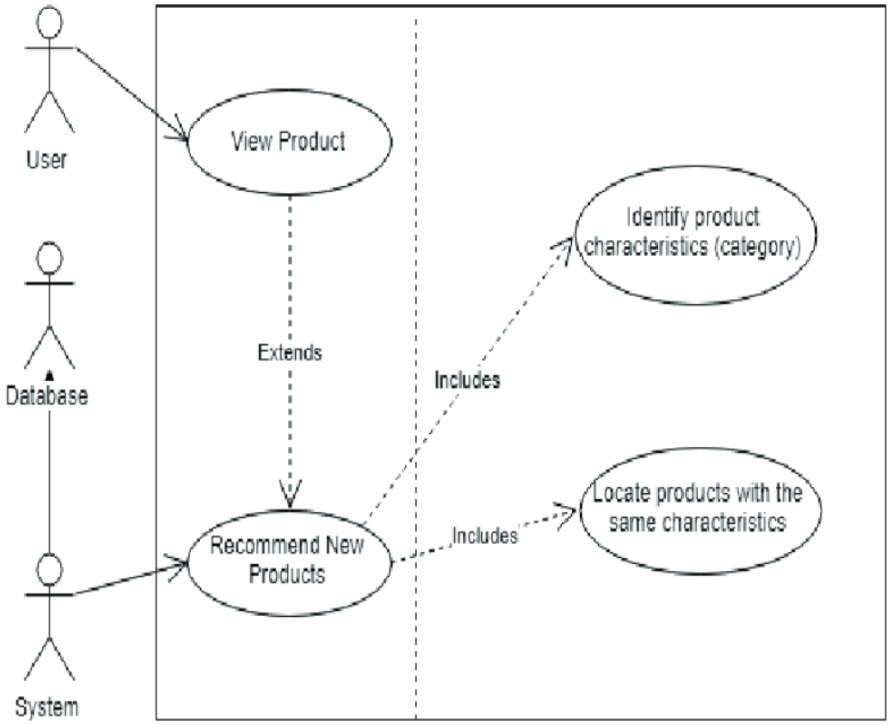

Figure 4.3: **Use case diagram for Personalized Medicine Recommendation System**

The above Figure 4.3 represents the use case diagram for personalised medicine recommendation system in which the user i.e, the user accesses the website and enters the symptoms which gets processed by the machine learning model and the model returns the predicted output to the website and the website displays the result.

4.2.3 Class Diagram

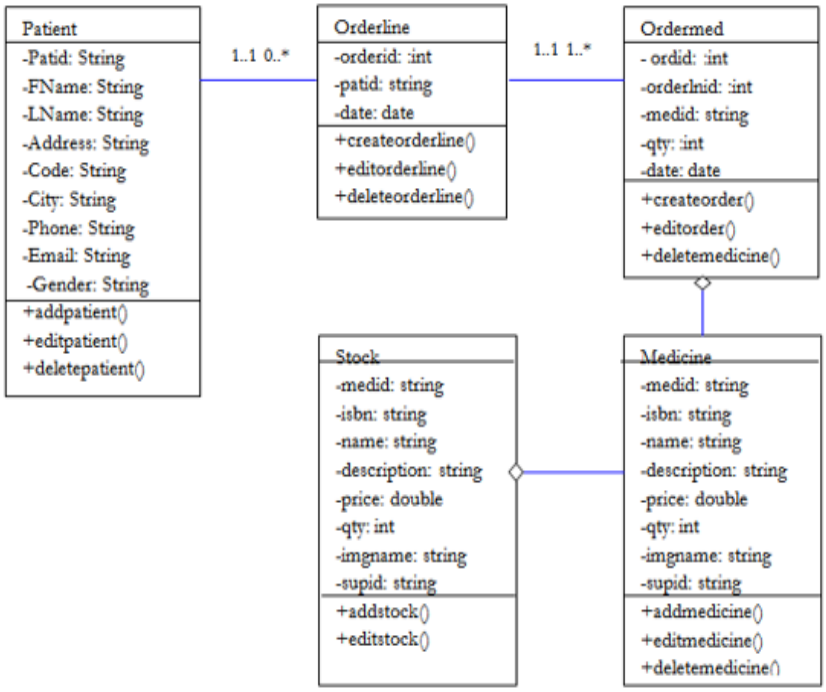

Figure 4.4: **Class Diagram for Personalized Medicine Recommendation using Machine Learning**

The above Figure 4.4 represents the class diagram of the Personalised Medicine recommendation system. It consists of the patient and the medicine classes. The Personalised Medicine recommendation system recommendation classes consists of patient, medicine, disease, symptoms and related related attributes along with the function prediction which implies to the predict class. The predict class contains the result function which displays the result.

4.2.4 Sequence Diagram

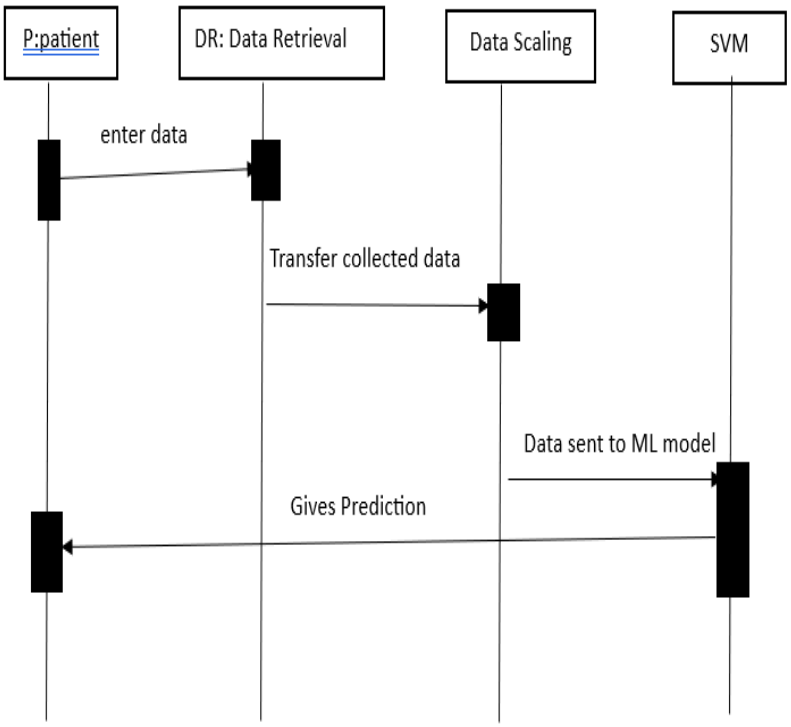

Figure 4.5: **Sequence diagram for Personalized Medicine Recommendation System**

The figure 4.5 tells about a sequence diagram that illustrates the chronological flow of interactions and messages exchanged between various components or modules within a system. In this, initially the patient gives the data to the website which gets interpreted by the backend program that is integrated with the help of flask framework. The data is scaled using standard and min-max scalar methods. After that, the support vector classifier processes this scaled data as input and predicts an appropriate medicine.

4.2.5 Activity Diagram

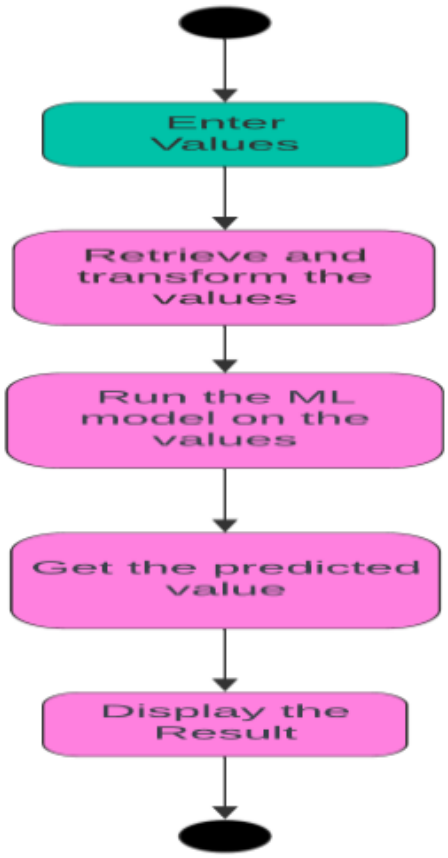

Figure 4.6: **Activity diagram for Personalized Medicine Recommendation System**

The figure 4.6 shows the activities done in the personalized medicine recommendation system. Initially, the user enters the values which gets retrieved and processed by the standard and min-max scalars. The processed data then enters the support vector classifier and returns a prediction. The predicted value then gets displayed on the webpage for the user.

4.3 Algorithm & Pseudo Code

4.3.1 Support Vector Machine(Classification)

Input Collection: Gather input data (medicines, symptoms) from the form submitted by the user.
Feature Transformation: Reshape the input data into a numpy array. Scale the features using the MinMaxScaler and StandardScaler. Combine the scaled features into

a final feature array.
Medicine Prediction: Use the pretrained machine learning model (Support Vector Classifier) to predict the medicine based on the final features.
Result Mapping: Map the numerical prediction of symptoms dataset to the medicine dictionary
Display Result: Render the result in the template, displaying the recommended medicine.

4.3.2 Pseudo Code

```
from flask import Flask, request, render_template, jsonify  # Import jsonify
import numpy as np
import pandas as pd
import pickle


# flask app
app = Flask(__name__)


# load model=========================================
svc = pickle.load(open('models/svc.pkl','rb'))
# creating routes========================================
@app.route("/")
def index():
    return render_template("index.html")

# Define a route for the home page
@app.route('/predict', methods=['GET', 'POST'])
predicted_disease=predicted_disease, dis_des=dis_des,
                                  my_precautions=my_precautions, medications=medications, my_diet=
                                      rec_diet,
                                  workout=workout)

    return render_template('index.html')



# about view funtion and path
@app.route('/about')
def about():
    return render_template("about.html")
# contact view funtion and path
@app.route('/contact')
def contact():
    return render_template("contact.html")
```

```
36
37 # developer view funtion and path
38 @app.route('/developer')
39 def developer():
40     return render_template("developer.html")
41
42 # about view funtion and path
43 @app.route('/blog')
44 def blog():
45     return render_template("blog.html")
46
47
48 if __name__ == '__main__':
49
50     app.run(debug=True)
```

4.4 Module Description

4.4.1 Data Collection and Preprocessing Module

The data collection module of a personalized medicine recommendation system serves as the foundation for generating tailored treatment recommendations. It encompasses various processes for gathering diverse datasets crucial for analyzing patient characteristics, genetic profiles, and relevant medical literature. The module integrates data acquisition from multiple sources, including electronic health records (EHRs), genetic databases, clinical trial repositories, and scientific literature databases. Upon data retrieval, preprocessing steps are applied to clean, standardize, and harmonize the collected data. This involves tasks such as missing value imputation, outlier detection, feature encoding, and normalization. Furthermore, specialized techniques are employed to handle the integration of heterogeneous data types, such as numerical, categorical, and textual data.

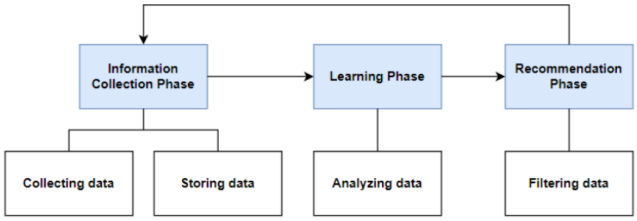

Figure 4.7: **Data collection and preprocessing structure**

4.4.2 Personalized medicine recommendation module

The personalized medicine recommendation module within a personalized medicine recommendation system serves as the core component responsible for analyzing patient-specific data and generating tailored treatment recommendations. This module employs advanced algorithms and machine learning techniques to interpret diverse datasets, including patient demographics, genetic profiles, medical history, and relevant clinical literature. By integrating these datasets, the module identifies correlations, associations, and predictive patterns that inform personalized treatment strategies. Leveraging patient-specific factors such as genetic variations, biomarkers, and treatment responses, the module generates recommendations tailored to individual patients' unique characteristics and medical needs.
The module is designed with scalability and adaptability in mind, capable of accommodating diverse data sources and evolving treatment paradigms. Furthermore, it offers a user-friendly interface for healthcare professionals to input patient data, visualize recommendations, and interpret model insights.Evaluation of the module is conducted using real-world clinical datasets, demonstrating its efficacy in providing accurate and interpretable personalized medicine recommendations.

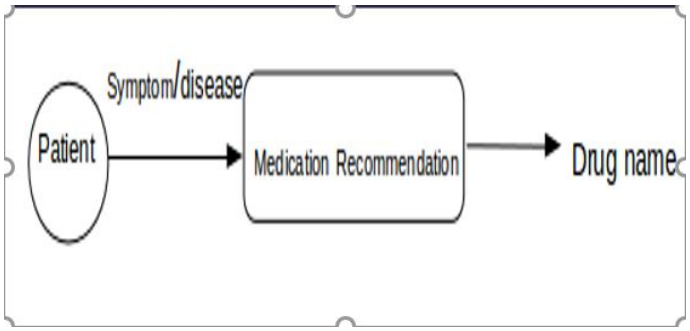

Figure 4.8: **Personalized recommendation structure**

4.4.3 Integration module

This module include data preprocessing pipelines for cleansing and harmonizing heterogeneous data sources, model deployment frameworks for hosting machine learning models in scalable and secure environments, and decision support interfaces for presenting personalized recommendations to healthcare providers in an intuitive and interpretable manner. It ensures the seamless integration of structured and unstructured data while maintaining data privacy and integrity. This module presents the design and implementation of an integration module for personalized medicine recom-

mendation using machine learning. The module serves as a bridge between disparate data sources, machine learning models, and clinical decision-making processes, facilitating the translation of predictive insights into actionable recommendations.

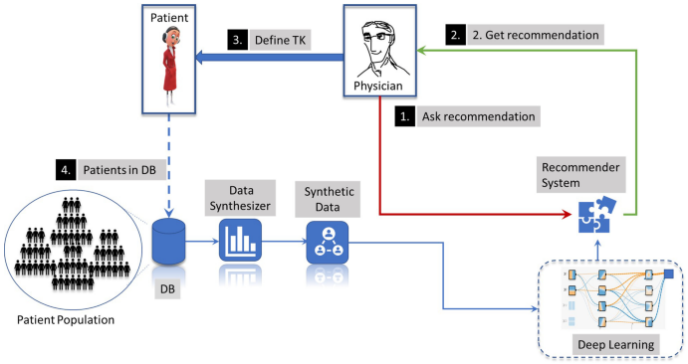

Figure 4.9: **Integration module Architecture**

4.4.4 User Interface module

The user interface module is designed to streamline the process of inputting patient data, obtaining personalized treatment recommendations, and visualizing the results in an interpretable manner. A structured form or interface for healthcare professionals to input patient information, including demographics, medical history, genetic data, and treatment preferences. The interface should support both manual data entry and integration with electronic health records (EHR) systems to facilitate seamless data retrieval. Clear and informative visualization of personalized treatment recommendations generated by the machine learning model. This may include graphical representations of predicted treatment outcomes, decision boundaries, and confidence intervals, allowing healthcare professionals to understand the rationale behind the recommendations.

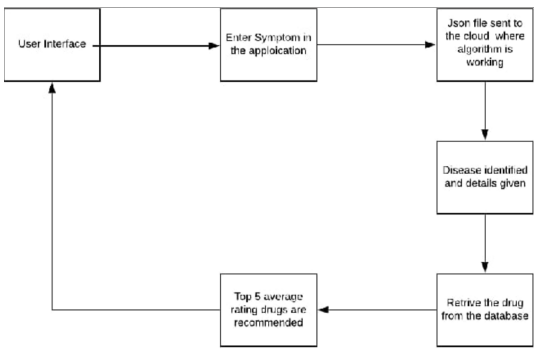

Figure 4.10: **User Interface module Architecture**

4.5 Steps to execute/run/implement the project

Data Collection

- Gather data on diseases types, symptoms and health conditions.

4.5.1 Data Preprocessing

- Clean and Preprocess the data to handle missing values, outliers, and inconsistencies.

4.5.2 Machine Learning Models

- Support Vector Machine (SVM) is a powerful machine learning technique that can be utilized within a personalized medicine recommendation system to predict treatment recommendations based on patient-specific data.
- The SVM algorithm works by finding the optimal hyperplane that best separates different classes within the data space. In the context of personalized medicine, this means identifying patterns or associations between patient features and treatment outcomes.

4.5.3 Integrate the Model with website

- Flask is a popular pyhton framework which helps in integrating the machine learning models with the website. It helps in routing the html pages with the backend program
- The data then processed through these models and the predicted value is displayed into the website using the jinja template.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

- Sections to input details about the patient's medical history, including past diagnoses, existing medical conditions, and any previous treatments or surgeries.
- Areas for describing the patient's current symptoms, complaints, or reasons for seeking medical advice.
- Areas for the patient to express their treatment preferences, goals, and concerns, ensuring a patient-centered approach to care.

5.1.2 Output Design

Recommendation Display:

- Clear and concise presentation of recommended medicine for disease symptoms.

5.2 Testing

5.3 Types of Testing

5.3.1 Unit testing

Input

```
import  pandas as pd
dataset = pd.read_csv('/content/Training.csv')
dataset
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
X = dataset.drop('prognosis', axis=1)
y = dataset['prognosis']
```

```
# ecoding prognonsis
le = LabelEncoder()
le.fit(y)
Y = le.transform(y)
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_state=20)
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, confusion_matrix
import numpy as np
# Create a dictionary to store models
models = {
    'SVC': SVC(kernel='linear'),
    'RandomForest': RandomForestClassifier(n_estimators=100, random_state=42),
    'GradientBoosting': GradientBoostingClassifier(n_estimators=100, random_state=42),
    'KNeighbors': KNeighborsClassifier(n_neighbors=5),
    'MultinomialNB': MultinomialNB()
}
# Loop through the models, train, test, and print results
for model_name, model in models.items():
    # Train the model
    model.fit(X_train, y_train)
    # Test the model
    predictions = model.predict(X_test)
    # Calculate accuracy
    accuracy = accuracy_score(y_test, predictions)
    print(f"{model_name} Accuracy: {accuracy}")
    # Calculate confusion matrix
    cm = confusion_matrix(y_test, predictions)
    print(f"{model_name} Accuracy: {accuracy}")
    print(f"{model_name} Confusion Matrix:")
    print(np.array2string(cm, separator=', '))
    # selecting svc
svc = SVC(kernel='linear')
svc.fit(X_train,y_train)
ypred = svc.predict(X_test)
accuracy_score(y_test,ypred)
# save svc
import pickle
pickle.dump(svc,open('svc.pkl','wb'))
# save svc
import pickle
pickle.dump(svc,open('svc.pkl','wb'))
# test 1:
print("predicted disease :",svc.predict(X_test.iloc[0].values.reshape(1,-1)))
print("Actual Disease :", y_test[0])
sym_des = pd.read_csv("symtoms_df.csv")
```

```
precautions = pd.read_csv("precautions_df.csv")
workout = pd.read_csv("workout_df.csv")
description = pd.read_csv("description.csv")
medications = pd.read_csv('medications.csv')
diets = pd.read_csv("diets.csv")
# Test 1
# Split the user's input into a list of symptoms (assuming they are comma-separated) # itching,
    skin_rash,nodal_skin_eruptions
symptoms = input("Enter your symptoms.....")
user_symptoms = [s.strip() for s in symptoms.split(',')]
# Remove any extra characters, if any
user_symptoms = [symptom.strip("[]' ") for symptom in user_symptoms]
predicted_disease = get_predicted_value(user_symptoms)

desc, pre, med, die, wrkout = helper(predicted_disease)

print("=================predicted disease============")
print(predicted_disease)
print("=================description==================")
print(desc)
print("=================precautions==================")
i = 1
for p_i in pre[0]:
    print(i, ": ", p_i)
    i += 1

print("=================medications==================")
for m_i in med:
    print(i, ": ", m_i)
    i += 1

print("=================workout======================")
for w_i in wrkout:
    print(i, ": ", w_i)
    i += 1

print("=================diets========================")
for d_i in die:
    print(i, ": ", d_i)
    i += 1
```

```
Enter your symptoms.......itching
==========predicted disease==========
Fungal infection
==========description==========
Fungal infection is a common skin condition caused by fungi.
==========precautions==========
1 :  bath twice
2 :  use detol or neem in bathing water
3 :  keep infected area dry
4 :  use clean cloths
==========medications==========
5 :  ['Antifungal Cream', 'Fluconazole', 'Terbinafine', 'Clotrimazole', 'Ketoconazole']
==========workout==========
6 :  Avoid sugary foods
7 :  Consume probiotics
8 :  Increase intake of garlic
9 :  Include yogurt in diet
10 :  Limit processed foods
11 :  Stay hydrated
12 :  Consume green tea
13 :  Eat foods rich in zinc
14 :  Include turmeric in diet
15 :  Eat fruits and vegetables
==========diets==========
16 :  ['Antifungal Diet', 'Probiotics', 'Garlic', 'Coconut oil', 'Turmeric']
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but SVC was fitted with feature n
  warnings.warn(
```

Figure 5.1: **Unit Testing Result**

5.3.2 Integration testing

Input

```python
from flask import Flask, request, render_template, jsonify  # Import jsonify
import numpy as np
import pandas as pd
import pickle


# flask app
app = Flask(__name__)



# load databasedataset===================================
sym_des = pd.read_csv("datasets/symtoms_df.csv")
precautions = pd.read_csv("datasets/precautions_df.csv")
workout = pd.read_csv("datasets/workout_df.csv")
description = pd.read_csv("datasets/description.csv")
medications = pd.read_csv('datasets/medications.csv')
diets = pd.read_csv("datasets/diets.csv")


# load model=========================================
svc = pickle.load(open('models/svc.pkl','rb'))
```

```
23
24
25 #=====================================================
26 # custome and helping functions
27 #==========================helper funtions===============
28 def helper(dis):
29     desc = description[description['Disease'] == dis]['Description']
30     desc = " ".join([w for w in desc])
31
32     pre = precautions[precautions['Disease'] == dis][['Precaution_1', 'Precaution_2', 'Precaution_3'
           , 'Precaution_4']]
33     pre = [col for col in pre.values]
34
35     med = medications[medications['Disease'] == dis]['Medication']
36     med = [med for med in med.values]
37
38     die = diets[diets['Disease'] == dis]['Diet']
39     die = [die for die in die.values]
40
41     wrkout = workout[workout['disease'] == dis] ['workout']
42
43
44     return desc,pre,med,die,wrkout
45
46 symptoms_dict = {'itching': 0, 'skin_rash': 1, 'nodal_skin_eruptions': 2, 'continuous_sneezing': 3,
     'shivering': 4, 'chills': 5, 'joint_pain': 6, 'stomach_pain': 7, 'acidity': 8, 'ulcers_on_tongue
     ': 9, 'muscle_wasting': 10, 'vomiting': 11, 'burning_micturition': 12, 'spotting_ urination':
     13, 'fatigue': 14, 'weight_gain': 15, 'anxiety': 16, 'cold_hands_and_feets': 17, 'mood_swings':
     18, 'weight_loss': 19, 'restlessness': 20, 'lethargy': 21, 'patches_in_throat': 22, '
     irregular_sugar_level': 23, 'cough': 24, 'high_fever': 25, 'sunken_eyes': 26, 'breathlessness':
     27, 'sweating': 28, 'dehydration': 29, 'indigestion': 30, 'headache': 31, 'yellowish_skin': 32,
     'dark_urine': 33, 'nausea': 34, 'loss_of_appetite': 35, 'pain_behind_the_eyes': 36, 'back_pain':
      37, 'constipation': 38, 'abdominal_pain': 39, 'diarrhoea': 40, 'mild_fever': 41, 'yellow_urine'
     : 42, 'yellowing_of_eyes': 43, 'acute_liver_failure': 44, 'fluid_overload': 45, '
     swelling_of_stomach': 46, 'swelled_lymph_nodes': 47, 'malaise': 48, '
     blurred_and_distorted_vision': 49, 'phlegm': 50, 'throat_irritation': 51, 'redness_of_eyes': 52,
      'sinus_pressure': 53, 'runny_nose': 54, 'congestion': 55, 'chest_pain': 56, 'weakness_in_limbs'
     : 57, 'fast_heart_rate': 58, 'pain_during_bowel_movements': 59, 'pain_in_anal_region': 60, '
     bloody_stool': 61, 'irritation_in_anus': 62, 'neck_pain': 63, 'dizziness': 64, 'cramps': 65, '
     bruising': 66, 'obesity': 67, 'swollen_legs': 68, 'swollen_blood_vessels': 69, '
     puffy_face_and_eyes': 70, 'enlarged_thyroid': 71, 'brittle_nails': 72, 'swollen_extremeties':
     73, 'excessive_hunger': 74, 'extra_marital_contacts': 75, 'drying_and_tingling_lips': 76, '
     slurred_speech': 77, 'knee_pain': 78, 'hip_joint_pain': 79, 'muscle_weakness': 80, 'stiff_neck':
      81, 'swelling_joints': 82, 'movement_stiffness': 83, 'spinning_movements': 84, 'loss_of_balance
     ': 85, 'unsteadiness': 86, 'weakness_of_one_body_side': 87, 'loss_of_smell': 88, '
     bladder_discomfort': 89, 'foul_smell_of urine': 90, 'continuous_feel_of_urine': 91, '
     passage_of_gases': 92, 'internal_itching': 93, 'toxic_look_(typhos)': 94, 'depression': 95, '
     irritability': 96, 'muscle_pain': 97, 'altered_sensorium': 98, 'red_spots_over_body': 99, '
     belly_pain': 100, 'abnormal_menstruation': 101, 'dischromic _patches': 102, 'watering_from_eyes'
     : 103, 'increased_appetite': 104, 'polyuria': 105, 'family_history': 106, 'mucoid_sputum': 107,
```

```
    'rusty_sputum': 108, 'lack_of_concentration': 109, 'visual_disturbances': 110, '
    receiving_blood_transfusion': 111, 'receiving_unsterile_injections': 112, 'coma': 113, '
    stomach_bleeding': 114, 'distention_of_abdomen': 115, 'history_of_alcohol_consumption': 116, '
    fluid_overload.1': 117, 'blood_in_sputum': 118, 'prominent_veins_on_calf': 119, 'palpitations':
    120, 'painful_walking': 121, 'pus_filled_pimples': 122, 'blackheads': 123, 'scurring': 124, '
    skin_peeling': 125, 'silver_like_dusting': 126, 'small_dents_in_nails': 127, 'inflammatory_nails
    ': 128, 'blister': 129, 'red_sore_around_nose': 130, 'yellow_crust_ooze': 131}
diseases_list = {15: 'Fungal infection', 4: 'Allergy', 16: 'GERD', 9: 'Chronic cholestasis', 14: '
    Drug Reaction', 33: 'Peptic ulcer diseae', 1: 'AIDS', 12: 'Diabetes ', 17: 'Gastroenteritis', 6:
     'Bronchial Asthma', 23: 'Hypertension ', 30: 'Migraine', 7: 'Cervical spondylosis', 32: '
    Paralysis (brain hemorrhage)', 28: 'Jaundice', 29: 'Malaria', 8: 'Chicken pox', 11: 'Dengue',
    37: 'Typhoid', 40: 'hepatitis A', 19: 'Hepatitis B', 20: 'Hepatitis C', 21: 'Hepatitis D', 22: '
    Hepatitis E', 3: 'Alcoholic hepatitis', 36: 'Tuberculosis', 10: 'Common Cold', 34: 'Pneumonia',
    13: 'Dimorphic hemmorhoids(piles)', 18: 'Heart attack', 39: 'Varicose veins', 26: '
    Hypothyroidism', 24: 'Hyperthyroidism', 25: 'Hypoglycemia', 31: 'Osteoarthristis', 5: 'Arthritis
    ', 0: '(vertigo) Paroymsal  Positional Vertigo', 2: 'Acne', 38: 'Urinary tract infection', 35: '
    Psoriasis', 27: 'Impetigo'}

# Model Prediction function
def get_predicted_value(patient_symptoms):
    input_vector = np.zeros(len(symptoms_dict))
    for item in patient_symptoms:
        input_vector[symptoms_dict[item]] = 1
    return diseases_list[svc.predict([input_vector])[0]]




# creating routes========================================


@app.route("/")
def index():
    return render_template("index.html")

# Define a route for the home page
@app.route('/predict', methods=['GET', 'POST'])
def home():
    if request.method == 'POST':
        symptoms = request.form.get('symptoms')
        # mysysms = request.form.get('mysysms')
        # print(mysysms)
        print(symptoms)
        if symptoms =="Symptoms":
            message = "Please either write symptoms or you have written misspelled symptoms"
            return render_template('index.html', message=message)
        else:

            # Split the user's input into a list of symptoms (assuming they are comma-separated)
            user_symptoms = [s.strip() for s in symptoms.split(',')]
```

```
            # Remove any extra characters, if any
            user_symptoms = [symptom.strip("[]' ") for symptom in user_symptoms]
            predicted_disease = get_predicted_value(user_symptoms)
            dis_des, precautions, medications, rec_diet, workout = helper(predicted_disease)

            my_precautions = []
            for i in precautions[0]:
                my_precautions.append(i)

            return render_template('index.html', predicted_disease=predicted_disease, dis_des=
                dis_des,
                                   my_precautions=my_precautions, medications=medications, my_diet=
                                       rec_diet,
                                   workout=workout)

    return render_template('index.html')



# about view funtion and path
@app.route('/about')
def about():
    return render_template("about.html")
# contact view funtion and path
@app.route('/contact')
def contact():
    return render_template("contact.html")

# developer view funtion and path
@app.route('/developer')
def developer():
    return render_template("developer.html")

# about view funtion and path
@app.route('/blog')
def blog():
    return render_template("blog.html")


if __name__ == '__main__':

    app.run(debug=True)
```

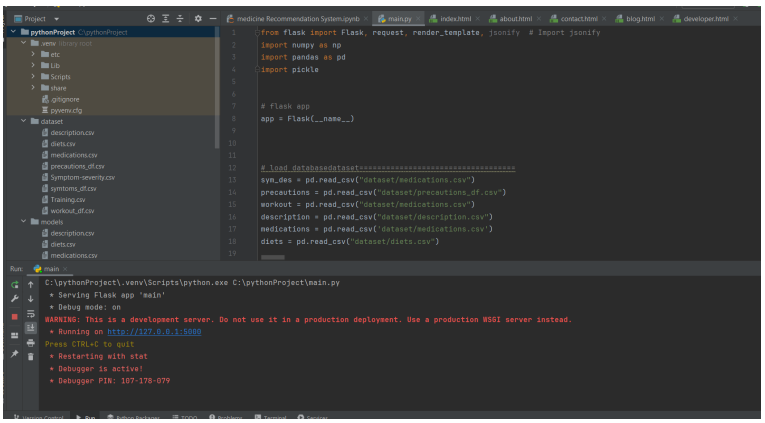

Figure 5.2: **Integration Testing Result**

Test result

5.3.3 System testing

Input

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Health Care Center</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="
        stylesheet" integrity="sha384-4bw+/aepP/YC94hEpVNVgiZdgIC5+VKNBQNGCHeKRQN+
        PtmoHDEXuppvnDJzQIu9" crossorigin="anonymous">
  </head>

  <style>

        .logo {
            width: 50px;
            height: 50px;
            color: black;
            margin-top: 0;
            margin-left: 2px;
        }

        .myimg {
            width: 50px;
            height: 50px;
            border: 2px solid black;
            border-radius: 25px;
        }


```

```
28
29
30      </style>
31  </head>
32  <body>
33      <!-- Navbar -->
34      <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
35          <div class="container-fluid">
36              <!-- Logo at the top-left corner -->
37              <div class="logo">
38                  <img class="myimg" src="{{ url_for('static', filename='img.png') }}" alt="">
39              </div>
40
41              <a class="navbar-brand" href="#">Health Center</a>
42              <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#
                    navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false"
                     aria-label="Toggle navigation">
43                  <span class="navbar-toggler-icon"></span>
44              </button>
45              <div class="collapse navbar-collapse" id="navbarSupportedContent">
46                  <ul class="navbar-nav me-auto mb-2 mb-lg-0">
47                      <li class="nav-item">
48                          <a class="nav-link active" aria-current="page" href="#">Home</a>
49                      </li>
50                      <li class="nav-item">
51                          <a class="nav-link" href="/about">About</a>
52                      </li>
53                      <li class="nav-item">
54                          <a class="nav-link" href="/contact">Contact</a>
55                      </li>
56                      <li class="nav-item">
57                          <a class="nav-link" href="/developer">Developer</a>
58                      </li>
59                      <li class="nav-item">
60                          <a class="nav-link" href="/blog">Blog</a>
61                      </li>
62                  </ul>
63                  <form class="d-flex" role="search">
64                      <input class="form-control me-2" type="search" placeholder="Search" aria-label="
                            Search">
65                      <button class="btn btn-outline-success" type="submit">Search</button>
66                  </form>
67              </div>
68          </div>
69      </nav>
70
71
72
73
74  <!-- main form of page -->
```

```
<h1 class="mt-4 my-4 text-center text-green">Health Care Center</h1>
<div class="container my-4 mt-4" style="background: black; color: white; border-radius: 15px;
    padding: 40px;">
    <form action="/predict" method="post">
        <div class="form-group">
            <label for="symptoms">Select Symptoms:</label>
            <input type="text" class="form-control", id="symptoms" name="symptoms" placeholder="type
                systems such as itching, sleeping, aching etc">

        </div>
        <br>
        <button type="button" id="startSpeechRecognition" class="btn btn-primary" style="margin-left
            :3px;border:1px solid white; border-radius:20px;">
            Start Speech Recognition
        </button>
        <br>

        <!-- Display the transcribed text here -->
        <div name="mysysms" id="transcription"></div>

        {% if message %}
        <p>{{ message }}</p>
        {% endif %}
        <br>

        <button type="submit" class="btn btn-danger btn-lg" style="width: 100%; padding: 14px;
            margin-bottom: 5px;">Predict</button>
    </form>
</div>









{% if predicted_disease %}

<!-- Results -->
<h1 class="text-center my-4 mt-4">Our AI System Results</h1>
<div class="container">

    <div class="result-container">
        <!-- Buttons to toggle display -->
        <button class="toggle-button" data-bs-toggle="modal" data-bs-target="#diseaseModal" style="
            padding:4px;  margin: 5px 40px 5px 0; font-size:20px;font-weight:bold; width:140px;
            border-radius:5px; background:#F39334;color:black;">Disease</button>
        <button class="toggle-button" data-bs-toggle="modal" data-bs-target="#descriptionModal"
            style="padding:4px; margin: 5px 40px 5px 0; font-size:20px;font-weight:bold; width:140px
```

```
            ; border-radius:5px; background:#268AF3 ;color:black;">Description</button>
        <button class="toggle-button" data-bs-toggle="modal" data-bs-target="#precautionModal" style
            ="padding:4px; margin: 5px 40px 5px 0; font-size:20px;font-weight:bold; width:140px;
            border-radius:5px; background:#F371F9 ;color:black;">Precaution</button>
        <button class="toggle-button" data-bs-toggle="modal" data-bs-target="#medicationsModal"
            style="padding:4px; margin: 5px 40px 5px 0; font-size:20px;font-weight:bold; width:140px
            ;border-radius:5px; background:#F8576F ;color:black;">Medications</button>
        <button class="toggle-button" data-bs-toggle="modal" data-bs-target="#workoutsModal" style="
            padding:4px; margin: 5px 40px 5px 0; font-size:20px;font-weight:bold; width:140px;
            border-radius:5px; background:#99F741 ;color:black;">Workouts</button>
        <button class="toggle-button" data-bs-toggle="modal" data-bs-target="#dietsModal" style="
            padding:4px; margin: 5px 40px 5px 0; font-size:20px;font-weight:bold; width:140px;
            border-radius:5px; background:#E5E23D;color:black;">Diets</button>
    </div>
</div>

{% endif %}

<!-- Disease Modal -->
    <div class="modal fade" id="diseaseModal" tabindex="-1" aria-labelledby="diseaseModalLabel" aria
        -hidden="true">
        <div class="modal-dialog">
            <div class="modal-content">
                <div class="modal-header" style="background-color: #020606; color:white;"> <!-- Set
                    header background color inline -->
                    <h5 class="modal-title" id="diseaseModalLabel">Predicted Disease</h5>
                    <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="
                        Close"></button>
                </div>
                <div class="modal-body" style="background-color: #modal-body-color;"> <!-- Set modal
                     body background color inline -->
                    <p>{{ predicted_disease }}</p>
                </div>
            </div>
        </div>
    </div>


    <!-- Description Modal -->
    <div class="modal fade" id="descriptionModal" tabindex="-1" aria-labelledby="
        descriptionModalLabel" aria-hidden="true">
        <div class="modal-dialog">
            <div class="modal-content">
                <div class="modal-header" style="background-color: #020606; color:white;">
                    <h5 class="modal-title" id="descriptionModalLabel">Description</h5>
                    <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="
                        Close"></button>
                </div>
                <div class="modal-body">
                    <p>{{ dis_des }}</p>
```

```
                </div>
            </div>
        </div>
    </div>

<!-- Precaution Modal -->
    <div class="modal fade" id="precautionModal" tabindex="-1" aria-labelledby="precautionModalLabel
        " aria-hidden="true">
        <div class="modal-dialog">
            <div class="modal-content">
                <div class="modal-header" style="background-color: #020606; color:white;">
                    <h5 class="modal-title" id="precautionModalLabel">Precaution</h5>
                    <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="
                        Close"></button>
                </div>
                <div class="modal-body">
                    <ul>
                        {% for i in my_precautions %}
                            <li>{{ i }}</li>
                        {% endfor %}
                    </ul>
                </div>
            </div>
        </div>
    </div>
```

5.3.4 Test Result

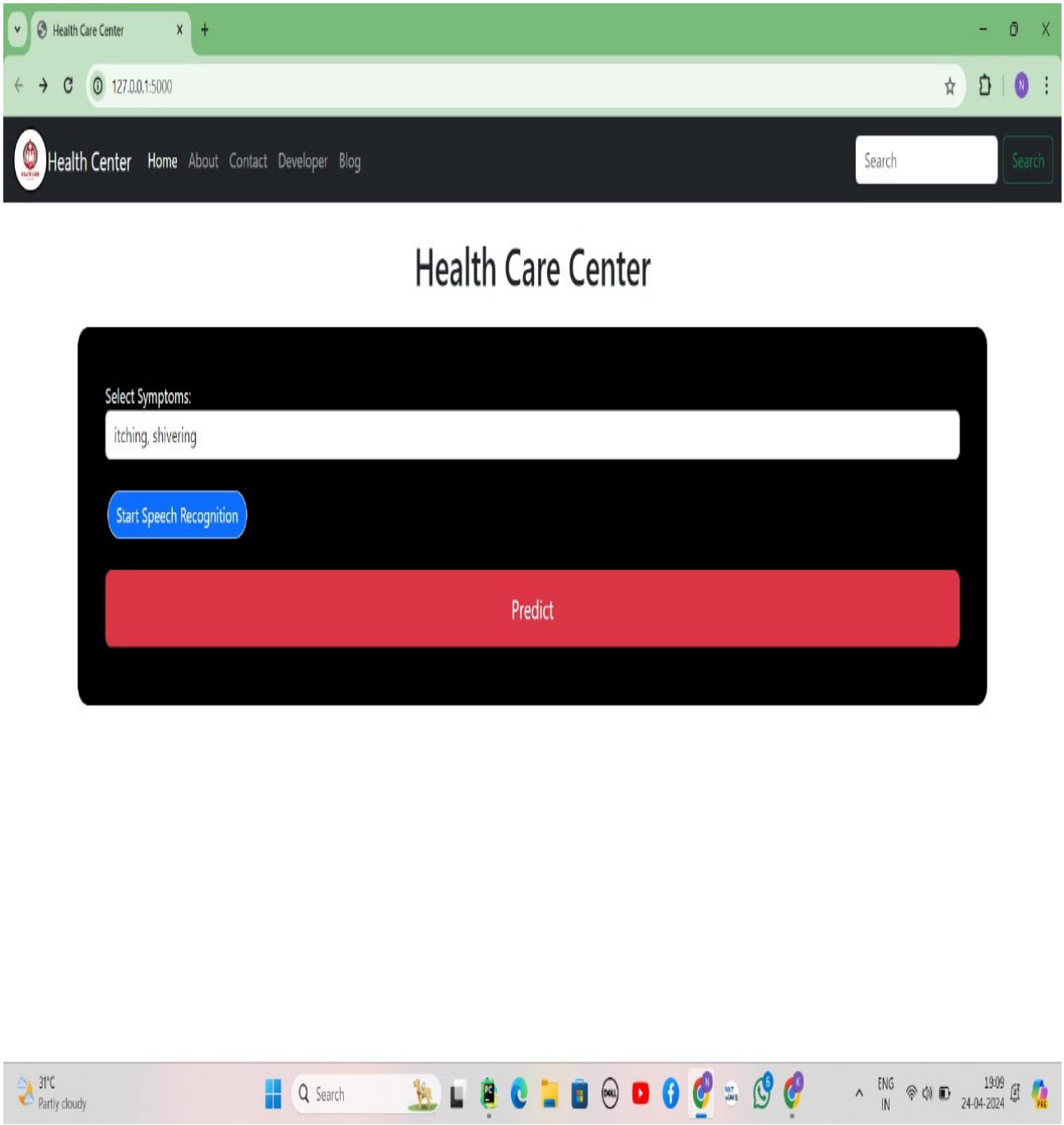

Figure 5.3: **Personalized Medicine Recommendation Test Image**

5.3.5 Test Result

```
Enter your symptoms.......itching
==========predicted disease==========
Fungal infection
==========description==========
Fungal infection is a common skin condition caused by fungi.
==========precautions==========
1 :  bath twice
2 :  use detol or neem in bathing water
3 :  keep infected area dry
4 :  use clean cloths
==========medications==========
5 :  ['Antifungal Cream', 'Fluconazole', 'Terbinafine', 'Clotrimazole', 'Ketoconazole']
==========workout==========
6 :  Avoid sugary foods
7 :  Consume probiotics
8 :  Increase intake of garlic
9 :  Include yogurt in diet
10 :  Limit processed foods
11 :  Stay hydrated
12 :  Consume green tea
13 :  Eat foods rich in zinc
14 :  Include turmeric in diet
15 :  Eat fruits and vegetables
==========diets==========
16 :  ['Antifungal Diet', 'Probiotics', 'Garlic', 'Coconut oil', 'Turmeric']
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but SVC was fitted with feature n
  warnings.warn(
```

Figure 5.4: **Output of recommended Medicine**

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The efficiency of the proposed personalized medicine recommendation system lies in its ability to swiftly analyze complex datasets and provide precise, tailored treatment recommendations based on individual patient characteristics and medical needs. By leveraging advanced algorithms, machine learning techniques, and comprehensive data integration, the system streamlines the decision-making process for healthcare providers. One aspect of efficiency is the system's speed in processing and analyzing vast amounts of patient data, genetic information, and medical literature. Utilizing optimized algorithms and scalable infrastructure, the system can swiftly extract relevant insights and generate personalized recommendations in real-time or near-real-time, enabling timely interventions and treatment adjustments. Moreover, the proposed system enhances efficiency through its accuracy in predicting treatment outcomes and identifying optimal interventions. By leveraging predictive modeling and data-driven insights, the system minimizes trial-and-error approaches to treatment, reducing the time and resources spent on ineffective therapies. This precision also mitigates the risk of adverse drug reactions and treatment failures, leading to improved patient outcomes and satisfaction.

6.2 Comparison of Existing and Proposed System

Existing system:(Decision tree)
In the Existing system, we implemented a decision tree algorithm that predicts medicines which are sometimes not applicable to disease symptoms. When using a decision tree model, it gives the training dataset the accuracy keeps improving with splits. We can easily overfit the dataset and doesn't know when it crossed the line unless we are using the cross validation. The advantages of the decision tree are model is very easy to interpret we can know that the variables and the value of the

variable is used to split the data. But the accuracy of decision tree in existing system gives less accurate output that is less when compared to proposed system.

Proposed system:(Support vector Machine algorithm)

Support Vector Machine is known for its effectiveness in handling high-dimensional data and capturing complex patterns within datasets. In personalized medicine, where patient data can be multifaceted and intricate, SVM's ability to delineate optimal hyperplanes for classification proves advantageous. SVM excels in scenarios where the relationships between patient features and treatment outcomes may not be linear and require sophisticated modeling. Its versatility in both classification and regression tasks makes it suitable for predicting categorical treatment decisions or continuous outcomes, such as drug response levels. Additionally, SVM offers robust generalization capabilities, allowing it to perform well on unseen patient cases, thus enhancing its applicability in personalized medicine recommendation systems.

6.3 Sample Code

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Health Care Center</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="
        stylesheet" integrity="sha384-4bw+/aepP/YC94hEpVNVgiZdgIC5+VKNBQNGCHeKRQN+
        PtmoHDEXuppvnDJzQIu9" crossorigin="anonymous">
  </head>

  <style>

        .logo {
            width: 50px;
            height: 50px;
            color: black;
            margin-top: 0;
            margin-left: 2px;
        }

        .myimg {
            width: 50px;
            height: 50px;
            border: 2px solid black;
            border-radius: 25px;
        }
```

```
26
27
28
29
30      </style>
31  </head>
32  <body>
33      <!-- Navbar -->
34      <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
35          <div class="container-fluid">
36              <!-- Logo at the top-left corner -->
37              <div class="logo">
38                  <img class="myimg" src="{{ url_for('static', filename='img.png') }}" alt="">
39              </div>
40
41              <a class="navbar-brand" href="#">Health Center</a>
42              <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#
                    navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false"
                     aria-label="Toggle navigation">
43                  <span class="navbar-toggler-icon"></span>
44              </button>
45              <div class="collapse navbar-collapse" id="navbarSupportedContent">
46                  <ul class="navbar-nav me-auto mb-2 mb-lg-0">
47                      <li class="nav-item">
48                          <a class="nav-link active" aria-current="page" href="#">Home</a>
49                      </li>
50                      <li class="nav-item">
51                          <a class="nav-link" href="/about">About</a>
52                      </li>
53                      <li class="nav-item">
54                          <a class="nav-link" href="/contact">Contact</a>
55                      </li>
56                      <li class="nav-item">
57                          <a class="nav-link" href="/developer">Developer</a>
58                      </li>
59                      <li class="nav-item">
60                          <a class="nav-link" href="/blog">Blog</a>
61                      </li>
62                  </ul>
63                  <form class="d-flex" role="search">
64                      <input class="form-control me-2" type="search" placeholder="Search" aria-label="
                            Search">
65                      <button class="btn btn-outline-success" type="submit">Search</button>
66                  </form>
67              </div>
68          </div>
69      </nav>
70
71
72
```

```
73
74 <!-- main form of page -->
75 <h1 class="mt-4 my-4 text-center text-green">Health Care Center</h1>
76 <div class="container my-4 mt-4" style="background: black; color: white; border-radius: 15px;
       padding: 40px;">
77     <form action="/predict" method="post">
78         <div class="form-group">
79             <label for="symptoms">Select Symptoms:</label>
80             <input type="text" class="form-control", id="symptoms" name="symptoms" placeholder="type
                   systems such as itching, sleeping, aching etc">
81
82         </div>
83         <br>
84         <button type="button" id="startSpeechRecognition" class="btn btn-primary" style="margin-left
               :3px;border:1px solid white; border-radius:20px;">
85             Start Speech Recognition
86         </button>
87         <br>
88
89         <!-- Display the transcribed text here -->
90         <div name="mysysms" id="transcription"></div>
91
92         {% if message %}
93         <p>{{ message }}</p>
94         {% endif %}
95         <br>
96
97         <button type="submit" class="btn btn-danger btn-lg" style="width: 100%; padding: 14px;
               margin-bottom: 5px;">Predict</button>
98     </form>
99 </div>
100
101
102
103
104
105
106
107
108 {% if predicted_disease %}
109
110 <!-- Results -->
111 <h1 class="text-center my-4 mt-4">Our AI System Results</h1>
112 <div class="container">
113
114     <div class="result-container">
115         <!-- Buttons to toggle display -->
116         <button class="toggle-button" data-bs-toggle="modal" data-bs-target="#diseaseModal" style="
               padding:4px;  margin: 5px 40px 5px 0; font-size:20px;font-weight:bold; width:140px;
               border-radius:5px; background:#F39334;color:black;">Disease</button>
```

```
117         <button class="toggle-button" data-bs-toggle="modal" data-bs-target="#descriptionModal"
                style="padding:4px; margin: 5px 40px 5px 0; font-size:20px;font-weight:bold; width:140px
                ; border-radius:5px; background:#268AF3 ;color:black;">Description</button>
118         <button class="toggle-button" data-bs-toggle="modal" data-bs-target="#precautionModal" style
                ="padding:4px; margin: 5px 40px 5px 0; font-size:20px;font-weight:bold; width:140px;
                border-radius:5px; background:#F371F9 ;color:black;">Precaution</button>
119         <button class="toggle-button" data-bs-toggle="modal" data-bs-target="#medicationsModal"
                style="padding:4px; margin: 5px 40px 5px 0; font-size:20px;font-weight:bold; width:140px
                ;border-radius:5px; background:#F8576F ;color:black;">Medications</button>
120         <button class="toggle-button" data-bs-toggle="modal" data-bs-target="#workoutsModal" style="
                padding:4px; margin: 5px 40px 5px 0; font-size:20px;font-weight:bold; width:140px;
                border-radius:5px; background:#99F741 ;color:black;">Workouts</button>
121         <button class="toggle-button" data-bs-toggle="modal" data-bs-target="#dietsModal" style="
                padding:4px; margin: 5px 40px 5px 0; font-size:20px;font-weight:bold; width:140px;
                border-radius:5px; background:#E5E23D;color:black;">Diets</button>
122     </div>
123 </div>
124
125 {% endif %}
126
127 <!-- Disease Modal -->
128     <div class="modal fade" id="diseaseModal" tabindex="-1" aria-labelledby="diseaseModalLabel" aria
            -hidden="true">
129         <div class="modal-dialog">
130             <div class="modal-content">
131                 <div class="modal-header" style="background-color: #020606; color:white;"> <!-- Set
                        header background color inline -->
132                     <h5 class="modal-title" id="diseaseModalLabel">Predicted Disease</h5>
133                     <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="
                            Close"></button>
134                 </div>
135                 <div class="modal-body" style="background-color: #modal-body-color;"> <!-- Set modal
                        body background color inline -->
136                     <p>{{ predicted_disease }}</p>
137                 </div>
138             </div>
139         </div>
140     </div>
141
142
143     <!-- Description Modal -->
144     <div class="modal fade" id="descriptionModal" tabindex="-1" aria-labelledby="
            descriptionModalLabel" aria-hidden="true">
145         <div class="modal-dialog">
146             <div class="modal-content">
147                 <div class="modal-header" style="background-color: #020606; color:white;">
148                     <h5 class="modal-title" id="descriptionModalLabel">Description</h5>
149                     <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="
                            Close"></button>
150                 </div>
```

```
                <div class="modal-body">
                    <p>{{ dis_des }}</p>
                </div>
            </div>
        </div>
    </div>

<!-- Precaution Modal -->
    <div class="modal fade" id="precautionModal" tabindex="-1" aria-labelledby="precautionModalLabel
        " aria-hidden="true">
        <div class="modal-dialog">
            <div class="modal-content">
                <div class="modal-header" style="background-color: #020606; color:white;">
                    <h5 class="modal-title" id="precautionModalLabel">Precaution</h5>
                    <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="
                        Close"></button>
                </div>
                <div class="modal-body">
                    <ul>
                        {% for i in my_precautions %}
                            <li>{{ i }}</li>
                        {% endfor %}
                    </ul>
                </div>
            </div>
        </div>
    </div>



    <!-- Medications Modal -->
    <div class="modal fade" id="medicationsModal" tabindex="-1" aria-labelledby="
        medicationsModalLabel" aria-hidden="true">
        <div class="modal-dialog">
            <div class="modal-content">
                <div class="modal-header" style="background-color: #020606; color:white;">
                    <h5 class="modal-title" id="medicationsModalLabel">Medications</h5>
                    <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="
                        Close"></button>
                </div>
                <div class="modal-body">
                    <ul>
                        {% for i in medications %}
                            <li>{{ i }}</li>
                        {% endfor %}
                    </ul>
                </div>
            </div>
        </div>
```

```
    </div>

    <!-- Workouts Modal -->
    <div class="modal fade" id="workoutsModal" tabindex="-1" aria-labelledby="workoutsModalLabel"
        aria-hidden="true">
        <div class="modal-dialog" >
            <div class="modal-content">
                <div class="modal-header" style="background-color: #020606; color:white;">
                    <h5 class="modal-title" id="workoutsModalLabel">Workouts</h5>
                    <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="
                        Close"></button>
                </div>
                <div class="modal-body">
                    <ul>
                        {% for i in workout %}
                            <li>{{ i }}</li>
                        {% endfor %}
                    </ul>
                </div>
            </div>
        </div>
    </div>

    <!-- Diets Modal -->
    <div class="modal fade" id="dietsModal" tabindex="-1" aria-labelledby="dietsModalLabel" aria-
        hidden="true">
        <div class="modal-dialog">
            <div class="modal-content">
                <div class="modal-header" style="background-color: #020606; color:white;">
                    <h5 class="modal-title" id="dietsModalLabel">Diets</h5>
                    <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="
                        Close"></button>
                </div>
                <div class="modal-body">
                    <ul>
                        {% for i in my_diet %}
                            <li>{{ i }}</li>
                        {% endfor %}
                    </ul>
                </div>
            </div>
        </div>
    </div>



        <script>
    const startSpeechRecognitionButton = document.getElementById('startSpeechRecognition');
    const transcriptionDiv = document.getElementById('transcription');

```

```
    startSpeechRecognitionButton.addEventListener('click', startSpeechRecognition);

    function startSpeechRecognition() {
        const recognition = new webkitSpeechRecognition(); // Use webkitSpeechRecognition for
            compatibility

        recognition.lang = 'en-US'; // Set the language for recognition

        recognition.onresult = function (event) {
            const result = event.results[0][0].transcript;
            transcriptionDiv.textContent = result;
        };

        recognition.onend = function () {
            console.log('Speech recognition ended.');
        };

        recognition.start();
    }
</script>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/bootstrap.bundle.min.js"
        integrity="sha384-HwwvtgBNo3bZJJLYd8oVXjrBZt8cqVSpeBNS5n7C8IVInixGAoxmnlMuBnhbgrkm"
        crossorigin="anonymous"></script>
</body>

</html>
```

Output

```
Enter your symptoms.......itching
==========predicted disease============
Fungal infection
==========description=================
Fungal infection is a common skin condition caused by fungi.
==========precautions=================
1 :  bath twice
2 :  use detol or neem in bathing water
3 :  keep infected area dry
4 :  use clean cloths
==========medications=================
5 :  ['Antifungal Cream', 'Fluconazole', 'Terbinafine', 'Clotrimazole', 'Ketoconazole']
==========workout=================
6 :  Avoid sugary foods
7 :  Consume probiotics
8 :  Increase intake of garlic
9 :  Include yogurt in diet
10 :  Limit processed foods
11 :  Stay hydrated
12 :  Consume green tea
13 :  Eat foods rich in zinc
14 :  Include turmeric in diet
15 :  Eat fruits and vegetables
==========diets=================
16 :  ['Antifungal Diet', 'Probiotics', 'Garlic', 'Coconut oil', 'Turmeric']
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but SVC was fitted with feature n
  warnings.warn(
```

Figure 6.1: **Medicine Recommended Output**

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

In conclusion, the development and implementation of a personalized medicine recommendation system represent a significant stride towards revolutionizing healthcare delivery. By harnessing the power of advanced algorithms, machine learning techniques, and comprehensive data integration, such a system holds immense promise in tailoring treatment recommendations to individual patient characteristics and medical needs. Through the synthesis of diverse datasets including patient demographics, genetic profiles, medical history, and relevant clinical literature, personalized medicine recommendation systems offer a pathway to precision healthcare delivery.The potential impact of personalized medicine recommendation systems is far-reaching. By providing accurate, timely, and tailored treatment recommendations, these systems have the capacity to optimize clinical decision-making, enhance patient outcomes, and improve the efficiency of healthcare delivery. Moreover, personalized medicine recommendation systems have the potential to drive innovation, foster interdisciplinary collaboration, and shape the future of healthcare by advancing precision medicine approaches. However, the realization of this potential hinges on addressing various challenges, including data privacy concerns, interoperability issues, and ensuring equitable access to personalized healthcare technologies

7.2 Future Enhancements

Future enhancements for medicine recommendation systems could focus on several key areas to further advance the capabilities and effectiveness of personalized healthcare delivery. Firstly, incorporating advanced artificial intelligence (AI) techniques such as deep learning and reinforcement learning could enhance the predic-

tive accuracy and robustness of recommendation models. These techniques have shown promise in handling complex, high-dimensional data and capturing intricate patterns and relationships within healthcare datasets. Furthermore, integrating real-time patient monitoring data from wearable devices and Internet of Things (IoT) sensors could enable dynamic, adaptive recommendations that respond to changes in patients' health status or treatment responses. By continuously capturing and analyzing patient-generated health data, recommendation systems could offer proactive interventions and personalized treatment adjustments in response to evolving patient needs. In addition, leveraging blockchain technology could enhance the security, transparency, and traceability of healthcare data, thereby addressing privacy concerns and facilitating secure data sharing and interoperability across healthcare systems. Blockchain-based solutions could enable patients to maintain ownership and control over their health data while still allowing authorized parties to access and contribute to their care.

Chapter 8

PLAGIARISM REPORT

Figure 8.1: **Plagiarism report**

Chapter 9

SOURCE CODE & POSTER PRESENTATION

9.1 Source Code

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Health Care Center</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="
        stylesheet" integrity="sha384-4bw+/aepP/YC94hEpVNVgiZdgIC5+VKNBQNGCHeKRQN+
        PtmoHDEXuppvnDJzQIu9" crossorigin="anonymous">
  </head>

  <style>

        .logo {
            width: 50px;
            height: 50px;
            color: black;
            margin-top: 0;
            margin-left: 2px;
        }

        .myimg {
            width: 50px;
            height: 50px;
            border: 2px solid black;
            border-radius: 25px;
        }




    </style>
</head>
<body>
    <!-- Navbar -->
```

```
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
        <div class="container-fluid">
            <!-- Logo at the top-left corner -->
            <div class="logo">
                <img class="myimg" src="{{ url_for('static', filename='img.png') }}" alt="">
            </div>

            <a class="navbar-brand" href="#">Health Center</a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#
                navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false"
                 aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                    <li class="nav-item">
                        <a class="nav-link active" aria-current="page" href="#">Home</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="/about">About</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="/contact">Contact</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="/developer">Developer</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="/blog">Blog</a>
                    </li>
                </ul>
                <form class="d-flex" role="search">
                    <input class="form-control me-2" type="search" placeholder="Search" aria-label="
                        Search">
                    <button class="btn btn-outline-success" type="submit">Search</button>
                </form>
            </div>
        </div>
    </nav>



<!-- main form of page -->
<h1 class="mt-4 my-4 text-center text-green">Health Care Center</h1>
<div class="container my-4 mt-4" style="background: black; color: white; border-radius: 15px;
    padding: 40px;">
    <form action="/predict" method="post">
        <div class="form-group">
            <label for="symptoms">Select Symptoms:</label>
```

```
80              <input type="text" class="form-control", id="symptoms" name="symptoms" placeholder="type
                    systems such as itching, sleeping, aching etc">
81
82          </div>
83          <br>
84          <button type="button" id="startSpeechRecognition" class="btn btn-primary" style="margin-left
                :3px;border:1px solid white; border-radius:20px;">
85              Start Speech Recognition
86          </button>
87          <br>
88
89          <!-- Display the transcribed text here -->
90          <div name="mysysms" id="transcription"></div>
91
92          {% if message %}
93          <p>{{ message }}</p>
94          {% endif %}
95          <br>
96
97          <button type="submit" class="btn btn-danger btn-lg" style="width: 100%; padding: 14px;
                margin-bottom: 5px;">Predict</button>
98      </form>
99  </div>
100
101
102
103
104
105
106
107
108 {% if predicted_disease %}
109
110 <!-- Results -->
111 <h1 class="text-center my-4 mt-4">Our AI System Results</h1>
112 <div class="container">
113
114     <div class="result-container">
115         <!-- Buttons to toggle display -->
116         <button class="toggle-button" data-bs-toggle="modal" data-bs-target="#diseaseModal" style="
                padding:4px;  margin: 5px 40px 5px 0; font-size:20px;font-weight:bold; width:140px;
                border-radius:5px; background:#F39334;color:black;">Disease</button>
117         <button class="toggle-button" data-bs-toggle="modal" data-bs-target="#descriptionModal"
                style="padding:4px; margin: 5px 40px 5px 0; font-size:20px;font-weight:bold; width:140px
                ; border-radius:5px; background:#268AF3 ;color:black;">Description</button>
118         <button class="toggle-button" data-bs-toggle="modal" data-bs-target="#precautionModal" style
                ="padding:4px; margin: 5px 40px 5px 0; font-size:20px;font-weight:bold; width:140px;
                border-radius:5px; background:#F371F9 ;color:black;">Precaution</button>
119         <button class="toggle-button" data-bs-toggle="modal" data-bs-target="#medicationsModal"
                style="padding:4px; margin: 5px 40px 5px 0; font-size:20px;font-weight:bold; width:140px
```

```
            ;border-radius:5px; background:#F8576F ;color:black;">Medications</button>
        <button class="toggle-button" data-bs-toggle="modal" data-bs-target="#workoutsModal" style="
            padding:4px; margin: 5px 40px 5px 0; font-size:20px;font-weight:bold; width:140px;
            border-radius:5px; background:#99F741 ;color:black;">Workouts</button>
        <button class="toggle-button" data-bs-toggle="modal" data-bs-target="#dietsModal" style="
            padding:4px; margin: 5px 40px 5px 0; font-size:20px;font-weight:bold; width:140px;
            border-radius:5px; background:#E5E23D;color:black;">Diets</button>
    </div>
</div>

{% endif %}

<!-- Disease Modal -->
    <div class="modal fade" id="diseaseModal" tabindex="-1" aria-labelledby="diseaseModalLabel" aria
        -hidden="true">
        <div class="modal-dialog">
            <div class="modal-content">
                <div class="modal-header" style="background-color: #020606; color:white;"> <!-- Set
                    header background color inline -->
                    <h5 class="modal-title" id="diseaseModalLabel">Predicted Disease</h5>
                    <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="
                        Close"></button>
                </div>
                <div class="modal-body" style="background-color: #modal-body-color;"> <!-- Set modal
                    body background color inline -->
                    <p>{{ predicted_disease }}</p>
                </div>
            </div>
        </div>
    </div>


    <!-- Description Modal -->
    <div class="modal fade" id="descriptionModal" tabindex="-1" aria-labelledby="
        descriptionModalLabel" aria-hidden="true">
        <div class="modal-dialog">
            <div class="modal-content">
                <div class="modal-header" style="background-color: #020606; color:white;">
                    <h5 class="modal-title" id="descriptionModalLabel">Description</h5>
                    <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="
                        Close"></button>
                </div>
                <div class="modal-body">
                    <p>{{ dis_des }}</p>
                </div>
            </div>
        </div>
    </div>

<!-- Precaution Modal -->
```

```
159     <div class="modal fade" id="precautionModal" tabindex="-1" aria-labelledby="precautionModalLabel
            " aria-hidden="true">
160         <div class="modal-dialog">
161             <div class="modal-content">
162                 <div class="modal-header" style="background-color: #020606; color:white;">
163                     <h5 class="modal-title" id="precautionModalLabel">Precaution</h5>
164                     <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="
                            Close"></button>
165                 </div>
166                 <div class="modal-body">
167                     <ul>
168                         {% for i in my_precautions %}
169                             <li>{{ i }}</li>
170                         {% endfor %}
171                     </ul>
172                 </div>
173             </div>
174         </div>
175     </div>
176
177
178
179
180     <!-- Medications Modal -->
181     <div class="modal fade" id="medicationsModal" tabindex="-1" aria-labelledby="
            medicationsModalLabel" aria-hidden="true">
182         <div class="modal-dialog">
183             <div class="modal-content">
184                 <div class="modal-header" style="background-color: #020606; color:white;">
185                     <h5 class="modal-title" id="medicationsModalLabel">Medications</h5>
186                     <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="
                            Close"></button>
187                 </div>
188                 <div class="modal-body">
189                     <ul>
190                         {% for i in medications %}
191                             <li>{{ i }}</li>
192                         {% endfor %}
193                     </ul>
194                 </div>
195             </div>
196         </div>
197     </div>
198
199     <!-- Workouts Modal -->
200     <div class="modal fade" id="workoutsModal" tabindex="-1" aria-labelledby="workoutsModalLabel"
            aria-hidden="true">
201         <div class="modal-dialog" >
202             <div class="modal-content">
203                 <div class="modal-header" style="background-color: #020606; color:white;">
```

```
                    <h5 class="modal-title" id="workoutsModalLabel">Workouts</h5>
                    <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="
                        Close"></button>
                </div>
                <div class="modal-body">
                    <ul>
                        {% for i in workout %}
                            <li>{{ i }}</li>
                        {% endfor %}
                    </ul>
                </div>
            </div>
        </div>
    </div>

    <!-- Diets Modal -->
    <div class="modal fade" id="dietsModal" tabindex="-1" aria-labelledby="dietsModalLabel" aria-
        hidden="true">
        <div class="modal-dialog">
            <div class="modal-content">
                <div class="modal-header" style="background-color: #020606; color:white;">
                    <h5 class="modal-title" id="dietsModalLabel">Diets</h5>
                    <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="
                        Close"></button>
                </div>
                <div class="modal-body">
                    <ul>
                        {% for i in my_diet %}
                            <li>{{ i }}</li>
                        {% endfor %}
                    </ul>
                </div>
            </div>
        </div>
    </div>



        <script>
    const startSpeechRecognitionButton = document.getElementById('startSpeechRecognition');
    const transcriptionDiv = document.getElementById('transcription');

    startSpeechRecognitionButton.addEventListener('click', startSpeechRecognition);

    function startSpeechRecognition() {
        const recognition = new webkitSpeechRecognition(); // Use webkitSpeechRecognition for
            compatibility

        recognition.lang = 'en-US'; // Set the language for recognition

```

```
250         recognition.onresult = function (event) {
251             const result = event.results[0][0].transcript;
252             transcriptionDiv.textContent = result;
253         };
254 
255         recognition.onend = function () {
256             console.log('Speech recognition ended.');
257         };
258 
259         recognition.start();
260     }
261 </script>
262 
263     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/bootstrap.bundle.min.js"
            integrity="sha384-HwwvtgBNo3bZJJLYd8oVXjrBZt8cqVSpeBNS5n7C8IVInixGAoxmnlMuBnhbgrkm"
            crossorigin="anonymous"></script>
264 </body>
265 
266 </html>
```

9.2 Poster Presentation

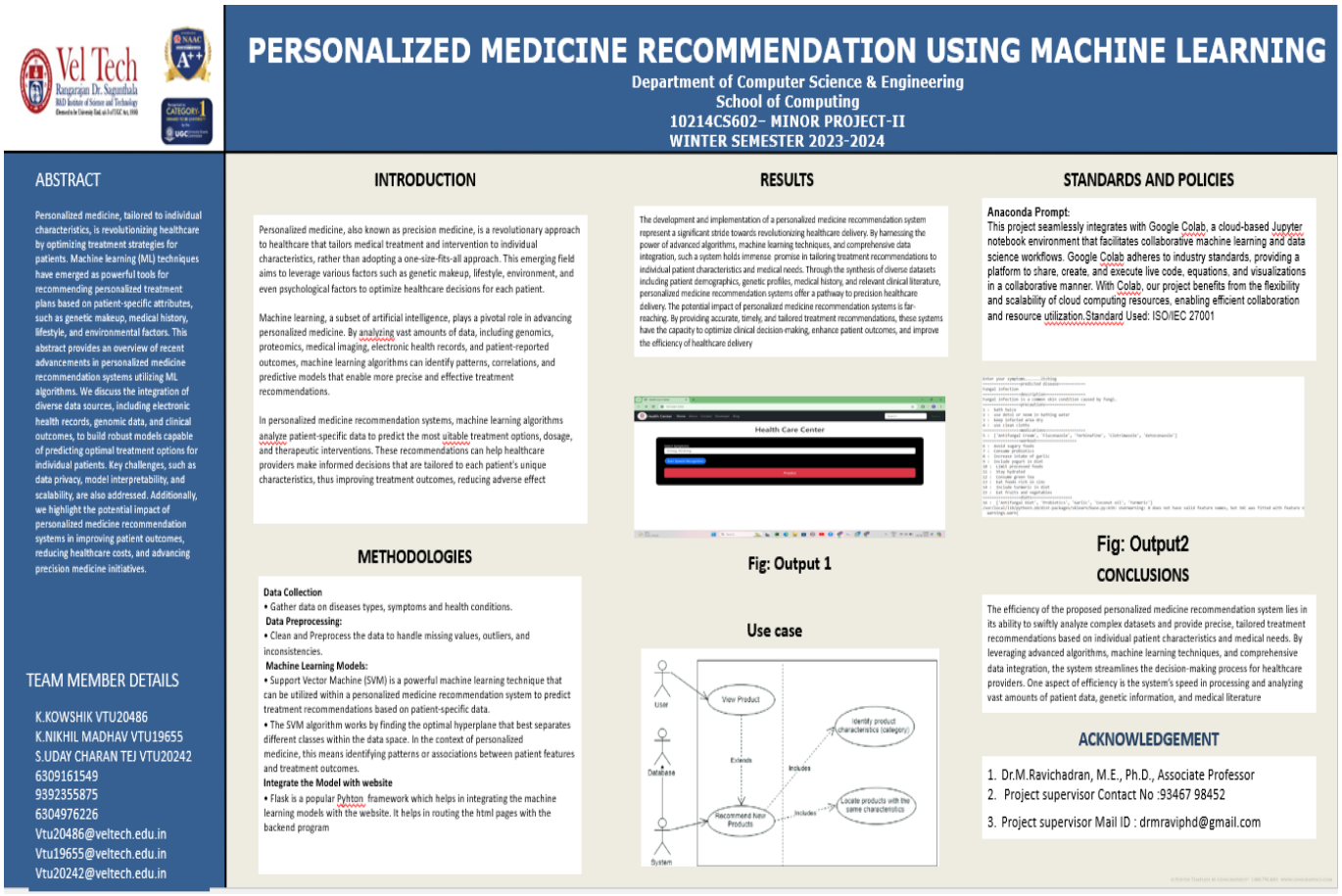

Figure 9.1: **Poster**

References

[1] R. Mu, "A survey of recommender systems based on deep learning," IEEE Access, vol. 6, pp. 69009–69022, 2023.
[2] A. Rajkomar, et al., "Scalable and accurate deep learning with electronic health records," IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 1, pp. 228-237, 2021.
[3] A. Esteva, et al., "A guide to deep learning in healthcare," Nature medicine, vol. 25, no. 1, pp. 24-29, 2020
[4]M. A. Alawad, et al., "Machine learning approach for the early diagnosis of schizophrenia patients," Journal of biomedical informatics, vol. 92, pp. 103142, 2021.
[5] S. Moonsamy, et al., "Enabling Precision Medicine through Individualized Pharmacokinetic Profiling," IEEE Transactions on Biomedical Engineering, vol. 66, no. 9, pp. 2644-2652, 2022..
[6] Y. Xie, et al., "A deep learning model integrating mutation and transcription features for cancer survival prediction," IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 9, pp. 2535-2545, 2021.
[7] S. H. Kim, et al., "A machine learning approach for the prediction of chronic obstructive pulmonary disease based on the blood eosinophil count and FEV1/FVC ratio," IEEE Access, vol. 8, pp. 117577-117586, 2020.
[8]H. H. Yang, et al., "Prediction of diabetes remission in morbidly obese patients after bariatric surgery using machine learning," IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 1, pp. 134-142, 2021.
[9] K. C. Yuen, et al., "Machine learning models for predicting hypoglycemia," Journal of Diabetes Science and Technology, vol. 12, no. 6, pp. 1235-1243, 2021.
[10] H. H. Yang, et al., "Machine learning models for predicting patients at risk for early hypoglycemia during admission," Diabetes research and clinical practice, vol. 158, pp. 107919, 2022.
[11] Y. Zhao, et al., "Risk prediction models for postoperative delirium: a systematic review and meta-analysis," Journal of critical care, vol. 50, pp. 1-10, 2021. [12] K. Wang, L. Liu, Z. Wu and Y. Zou, "Personalized Medicine Recommendation Using Collaborative Filtering and Deep Learn-

ing," in IEEE Access, vol. 7, pp. 18668-18676, 2020. [13] Y. Zhang, J. Wang, J. Jia, H. He and C. Zhang, "Personalized Medicine Recommendation Based on Medical Data and Deep Learning," 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Seoul, Korea (South), 2020, pp. 1128-1132.