iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: VII Month of publication: July 2025

DOI: https://doi.org/10.22214/ijraset.2025.73310

Symptoms Based Disease Identification Using SVC

Rongala Rajesh[1], Kuncha Lakshman[2]
Assistant Professor, MCA Final semester, Master of Computer Applications, Sanketika Vidya Parishad Engineering College, Vishakhapatnam, Andhra Pradesh, India

Abstract: This Flask-based application constitutes a comprehensive symptoms based disease identification platform designed to cater to a tripartite user base: patients, healthcare professionals (HCPs), and administrators. The core functionality revolves around a robust machine learning model capable of identifying potential diseases based on user-inputted symptoms. Patients can seamlessly input their symptoms, receive predicted disease diagnoses, and access accompanying information such as precautionary measures and recommended treatments. Healthcare professionals, on the other hand, are empowered to delve into patient data, refine predictive models, and offer specialized medical counsel. The administrative component ensures efficient management of HCP accounts and system oversight. Underpinning the application is a robust technological stack incorporating Flask for web development, pandas and NumPy for data manipulation, Joblib for model handling, and PyMongo for database interactions. The system leverages pre-trained models and CSV datasets for enhanced performance. Jinja2 facilitates dynamic template rendering, contributing to a user-friendly interface. This integrated platform seeks to revolutionize disease management by providing an efficient, accurate, and accessible tool for both patients and healthcare providers. The application serves as a comprehensive disease identification platform designed to augment healthcare professionals' decision-making capabilities. Recognizing the immense pressures healthcare providers face, this platform offers a tool to enhance diagnostic accuracy and efficiency. By leveraging a sophisticated machine learning model, the system empowers clinicians to input patient symptoms and receive potential disease diagnoses, thereby expediting the diagnostic process. This rapid analysis can be particularly beneficial in time-critical situations, such as emergency departments, where swift and accurate diagnoses are paramount. Moreover, the platform provides healthcare professionals with access to relevant medical information, including treatment guidelines, medication details, and potential complications, facilitating informed decision-making. By streamlining the diagnostic process and providing essential clinical data, this application aims to mitigate the effects of stress and burnout often experienced by healthcare professionals, ultimately contributing to improved patient outcomes and a more resilient healthcare workforce. The web application also consists of a feedback mechanism which allows health care professionals to send feedback on the working of the application on disease identification further increasing scope for improving performance and accuracy of the dataset and disease identification method.

Index Terms: Support Vector Classifier (SVC), Machine Learning Supervised Learning, Disease Prediction, Symptom-based Diagnosis, Sigmoid Kernel, Model Evaluation, Accuracy.

I. INTRODUCTION

The project "Symptoms Based Disease Identification Using SVC" is a web-based healthcare application designed to predict diseases based on user-inputted symptoms using machine learning. It addresses key challenges in the healthcare industry such as diagnostic delays, limited access to medical expertise, and overburdened healthcare systems. By leveraging a Support Vector Classifier (SVC), the system accurately classifies diseases from a dataset containing 132 symptoms and 40 diseases. The platform is built using Python with Flask for the backend and MongoDB for database management. Patients can enter their symptoms through a simple interface and receive disease predictions along with preventive measures, recommended medications, diet, and workout plans. The system features three roles: patients, healthcare professionals (HCPs), and administrators, each with specific access and functionality. HCPs can log in to validate and enhance predictions, while administrators manage user approvals and system settings. The application utilizes datasets in CSV format and a pre-trained model serialized using Joblib. It incorporates user authentication and role-based access control to ensure data security. Dynamic templates rendered with Jinja2 improve the user experience. The inclusion of HCP feedback helps refine the system for better accuracy over time. The platform also supports storing patient history and generating health records. This tool aims to reduce the cognitive burden on medical staff, particularly in high-pressure environments. It also empowers patients with early disease awareness and health education. Ultimately, the system enhances diagnostic efficiency, accessibility, and overall healthcare quality.

A. Existing System

In the existing healthcare system, disease diagnosis primarily relies on manual examination and the experience of healthcare professionals. This traditional approach involves a range of time-consuming processes such as symptom analysis, physical examination, and patient history evaluation, often without the aid of intelligent software tools[7]. Doctors and medical staff are burdened with heavy workloads, administrative responsibilities, and repetitive data entry tasks, which can lead to fatigue, increased stress, and human errors. The lack of digital tools limits real-time access to patient records and clinical guidelines, making evidence-based diagnosis and treatment decisions more challenging. Communication gaps among medical teams and between patients and providers further delay care. Moreover, the absence of an automated system means symptom tracking is subjective and prone to inconsistencies. As a result, patients may experience delayed or inaccurate diagnoses, while healthcare providers struggle with inefficiencies that hinder productivity and job satisfaction [13]. These limitations highlight the need for an intelligent, automated platform that can support clinicians and empower patients by providing accurate, data-driven disease predictions and health recommendations.

1) Challenges

a) Data Quality and Availability: Finding a comprehensive and clean dataset with labeled symptoms and corresponding diseases was difficult. Incomplete or inconsistent data can impact model accuracy.

b) Symptom Overlap Between Diseases: Many diseases share similar symptoms, making it challenging for the model to differentiate between them, especially for multi-class classification [12].

c) Model Overfitting: Some models showed high accuracy on training data but performed poorly on unseen data. Managing overfitting was essential to ensure real-world applicability.

d) Limited Dataset Size: With only 400 samples covering 40 diseases, the dataset lacked enough variation, which limits the generalizability of the model.

e) Input Standardization: Users may input symptoms with spelling errors or in different formats. Preprocessing and standardizing user input was crucial but complex [18].

f) Model Interpretability: While SVC provides good accuracy, it lacks interpretability, which is critical in healthcare where understanding the “why” behind a prediction is important.

B. Proposed System

The proposed system is an intelligent, web-based healthcare application that leverages machine learning to predict diseases based on user-inputted symptoms [1]. Designed using Python’s Flask framework, this system incorporates a Support Vector Classifier (SVC) model trained on a labeled dataset of 132 symptoms mapped to 40 diseases. The platform is built to serve three user roles—patients, healthcare professionals (HCPs), and administrators—each with role-specific access and functionality. Patients can input symptoms through a user-friendly interface and receive probable disease predictions, along with helpful details such as precautionary measures, medications, dietary suggestions, and exercise recommendations. Healthcare professionals can log in to view patient cases, validate diagnoses, and provide feedback to improve the model’s accuracy over time. The admin panel manages the approval and activation of HCP accounts, ensuring that only qualified personnel access the medical functionalities. The system uses MongoDB as the backend database to securely store user credentials, medical history, and prediction data. It also integrates preprocessed CSV files containing disease information for real-time display [19]. By automating the diagnostic process, reducing the cognitive burden on clinicians, and promoting preventive care, the proposed system aims to improve healthcare efficiency, accuracy, and accessibility—especially in settings with limited medical resources.

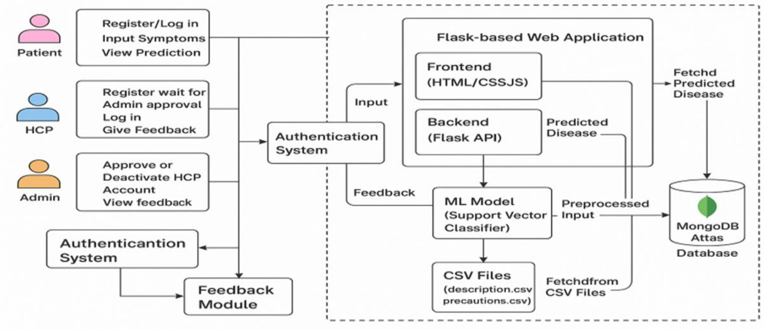


Fig: 1 Proposed Diagram

C. UML Diagrams

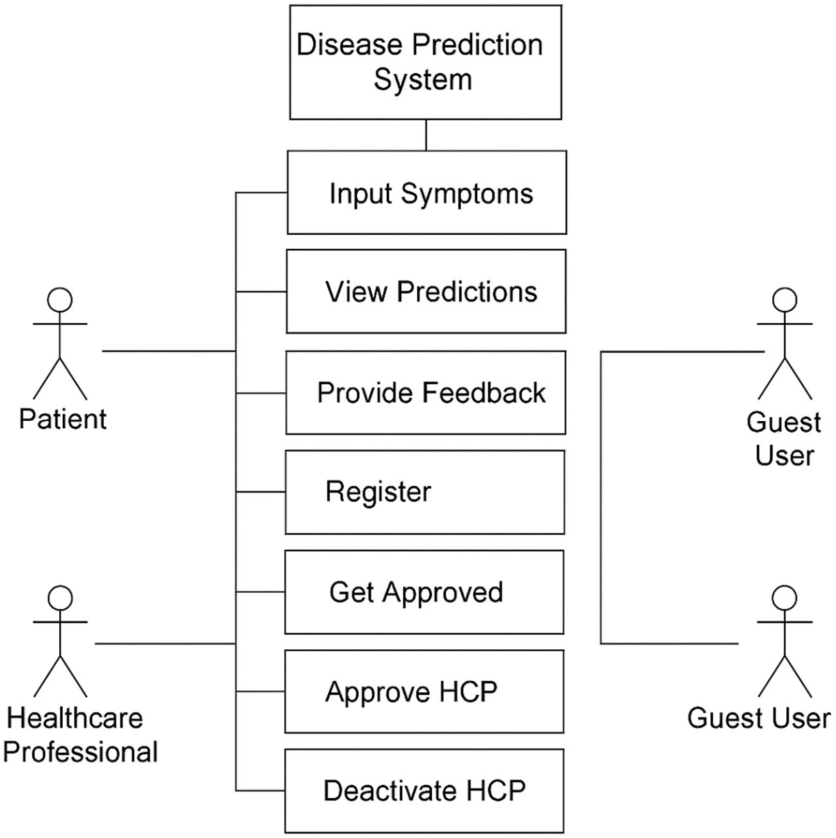


Fig: Usecase diagram

1) Advantages

Early Detection of Diseases

- The system can identify potential diseases based on early symptoms, helping in faster diagnosis and timely treatment [2].

Accurate Classification

- SVC is known for its high accuracy in binary and multiclass classification tasks, especially with non-linear and high-dimensional data.

Cost-Effective

- Reduces the need for expensive diagnostic tests in the initial stages by providing a preliminary diagnosis based on symptoms.

Assists Medical Practitioners

- Acts as a decision-support tool for doctors, improving their efficiency in diagnosing multiple conditions based on symptom inputs.

Scalable and Flexible

- The system can be scaled to include more diseases and symptoms, and updated as new medical data becomes available.

Handles Noisy and Incomplete Data

- SVC handles small and noisy datasets well, making it suitable for real-world medical data where symptoms may be vague or overlapping.

Non-Invasive

- Uses only symptom-based inputs, so its non-invasive and suitable for remote consultations or telemedicine applications [8].

Useful in Resource-Limited Areas

- Especially beneficial in rural or underdeveloped areas where access to specialists or diagnostic tools is limited.

II. ARCHITECTURE

The architecture of the Symptoms Based Disease Identification system using SVC (Support Vector Classifier) follows a structured and modular pipeline that begins with user input. The system first collects symptoms from users through a user interface, which can be a web form, mobile app, or chatbot [14]. These symptoms are then preprocessed using text cleaning, encoding, and normalization techniques to convert them into a machine-readable format. The cleaned data is passed to the feature extraction module, where key symptom patterns are transformed into numerical vectors suitable for classification. This processed data is then fed into the SVC model, which has been trained on a medical dataset mapping symptoms to diseases. Based on the input, the model predicts the most probable disease(s). The output is then passed to the result interpretation and display module, which shows the predicted disease, its confidence score, and potential next steps (e.g., seeing a doctor, recommended tests) [20]. The architecture may also include a feedback loop to improve model accuracy over time through user-confirmed results, making the system smarter with continued use.

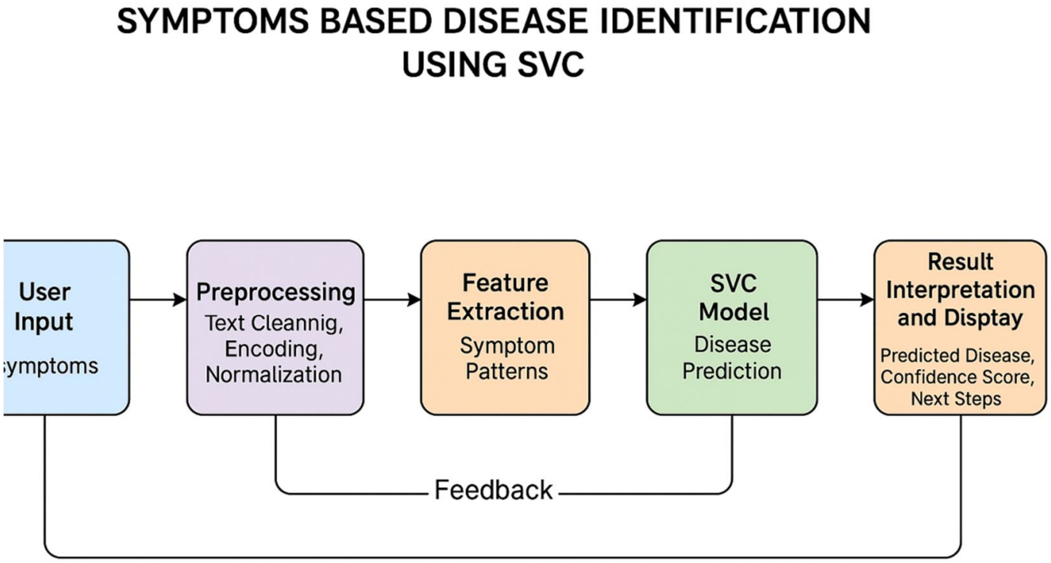

SYMPTOMS BASED DISEASE IDENTIFICATION USING SVC
User Input
ymptoms
Preprocessing
Text Cleannig, Encoding, Normalization
Feature Extraction
Symptom Patterns
SVC Model
Disease Prediction
Result Interpretation and Disptay
Predicted Disease, Confidence Score, Next Steps
Feedback

Fig:2 Architecture

A. Algorithm

Start

Collect the dataset containing symptoms as input features and corresponding diseases as labels.

Preprocess the data:

- Handle missing values.
- Encode categorical symptom values (e.g., One-Hot or Label Encoding).
- Normalize or scale features if needed.

Select important features (symptoms) using feature selection techniques [3].

Split the dataset into training and testing sets (e.g., 80% training, 20% testing).

Train the model:

- Use Support Vector Classifier (SVC) with appropriate kernel (linear, RBF, etc.).
- Fit the model on the training dataset.

Evaluate the model using metrics like accuracy, precision, recall, F1-score, and confusion matrix.

Take user input for symptoms (via form, GUI, or web interface).

Transform user input to match the model’s input format.

Use the trained SVC model to predict the most likely disease based on the symptoms.

Display the predicted disease to the user.

End

B. Techniques

The technique used in this project involves applying Support Vector Classification (SVC**)**, a supervised machine learning algorithm, to accurately identify diseases based on symptoms reported by users [9]. SVC works by finding the optimal hyperplane that separates different disease classes in a high-dimensional space formed by the symptom features. Initially, a dataset containing various symptoms and their corresponding diseases is collected and preprocessed. The symptoms are encoded into numerical form, and the dataset is split into training and testing sets. The SVC model is then trained on this data, learning the complex relationships between symptom patterns and disease labels. Once trained, the model can classify new symptom inputs provided by users and predict the most probable disease. To further enhance prediction accuracy, techniques such as feature selection and parameter tuning (e.g., using GridSearchCV) can be employed [15]. Additionally, the model's performance can be validated through cross-validation to ensure its generalizability across unseen data. The system can be integrated into a user-friendly interface that allows users to input symptoms easily, making it practical for real-world healthcare support. This technique is highly effective for multi-class classification problems and performs well even with limited data, making it suitable for medical decision-support systems based on symptom analysis.

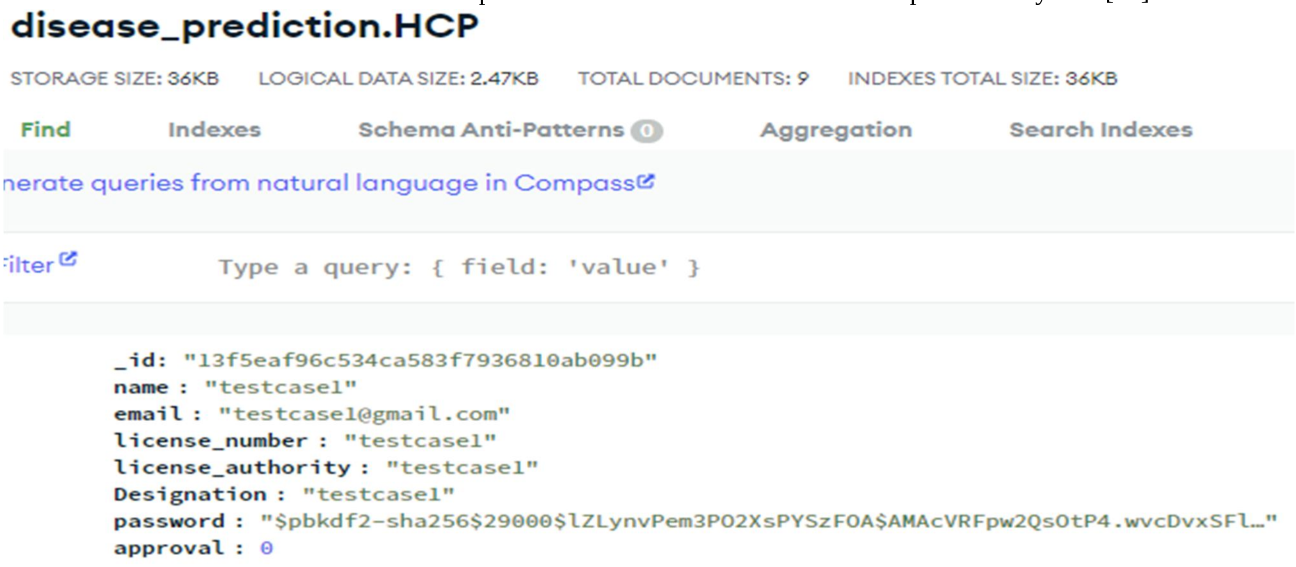
C. Tools

In the project "Symptoms Based Disease Identification Using SVC", various tools and technologies are employed to efficiently process data, build models, and deploy the system [21]. Python serves as the primary programming language due to its simplicity and wide range of libraries. For machine learning implementation, the Scikit-learn (sklearn) library is used to develop the Support Vector Classification (SVC) model. Pandas and NumPy are essential for handling and preprocessing the symptom dataset, while Matplotlib and Seaborn assist in visualizing the data and evaluating model performance. To build a user-friendly interface for interacting with the model [4], web frameworks like Flask or Streamlit can be utilized. Additionally, tools like Jupyter Notebook or Google Colab support interactive development and testing, making the overall system more robust and scalable. These tools collectively enable efficient symptom analysis and accurate disease prediction using SVC.

III. METHODOLOGY

A. Input

During the HCP (Healthcare Professional) registration process, the user is required to fill out a web form with specific personal and professional details. These include the full name, email address, designation, medical license number, and the authority that issued the license [10]. Additionally, the user must create a secure password to complete the signup. Once the form is submitted, the data is stored in a MongoDB collection named disease prediction. HCP. The password is securely hashed before storage to ensure user privacy and data protection. An approval field is also recorded with a default value of 0, indicating that the user is awaiting administrative verification before being granted access to the system. This process ensures that only verified healthcare professionals can access sensitive medical data or perform critical tasks within the disease prediction system [15].



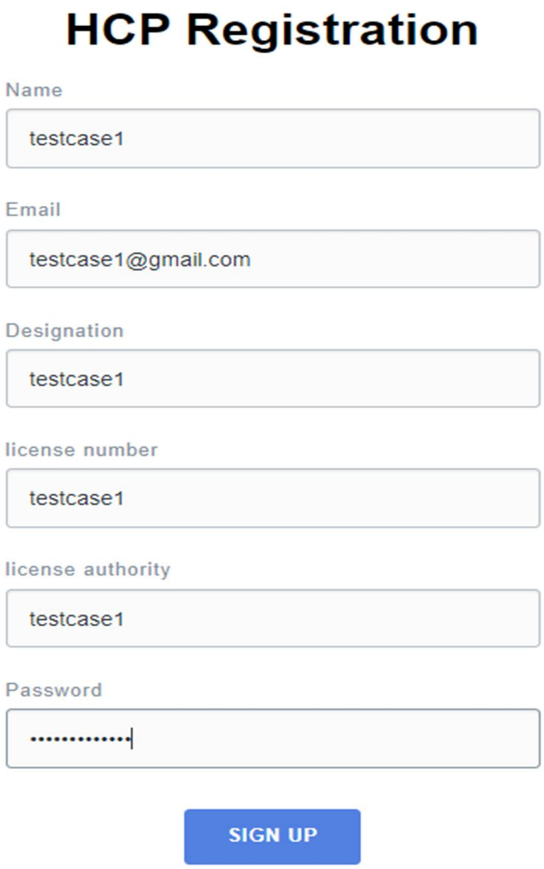
disease_prediction.HCP

STORAGE SIZE: 36KB   LOGICAL DATA SIZE: 2.47KB   TOTAL DOCUMENTS: 9   INDEXES TOTAL SIZE: 36KB

Find
Indexes
Schema Anti-Patterns 0
Aggregation
Search Indexes
nerate queries from natural language in Compass
Filter
Type a query: { field: 'value' }

```
_id: "13f5eaf96c534ca583f7936810ab099b"
name : "testcase1"
email : "testcase1@gmail.com"
license_number : "testcase1"
license_authority : "testcase1"
Designation : "testcase1"
password : "$pbkdf2-sha256$29000$lZLynvPem3PO2XsPYSzFOA$AMAcVRFpw2QsOtP4.wvcDvxSFl…"
approval : 0
```

Fig: testing data



HCP Registration

Name
testcase1

Email
testcase1@gmail.com

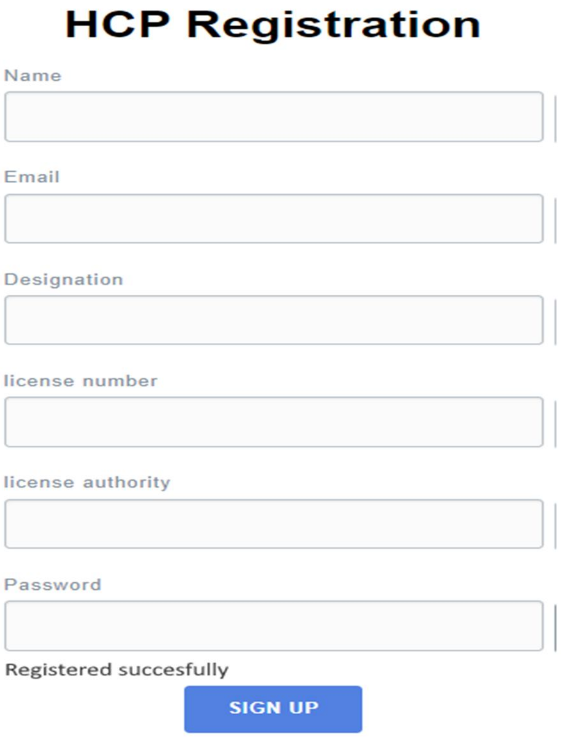
Designation
testcase1

license number
testcase1

license authority
testcase1

Password
•••••••••••••

Fig: checking data



HCP Registration

Name

Email

Designation

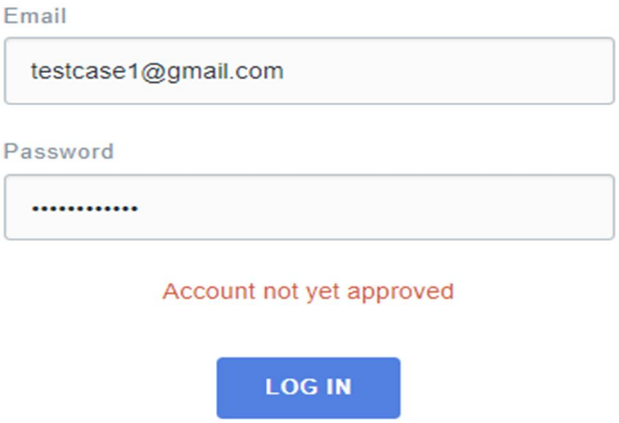
license number

license authority

Password

Registered succesfully

Fig: After Registration



Email

testcase1@gmail.com

Password

••••••••••••

Account not yet approved

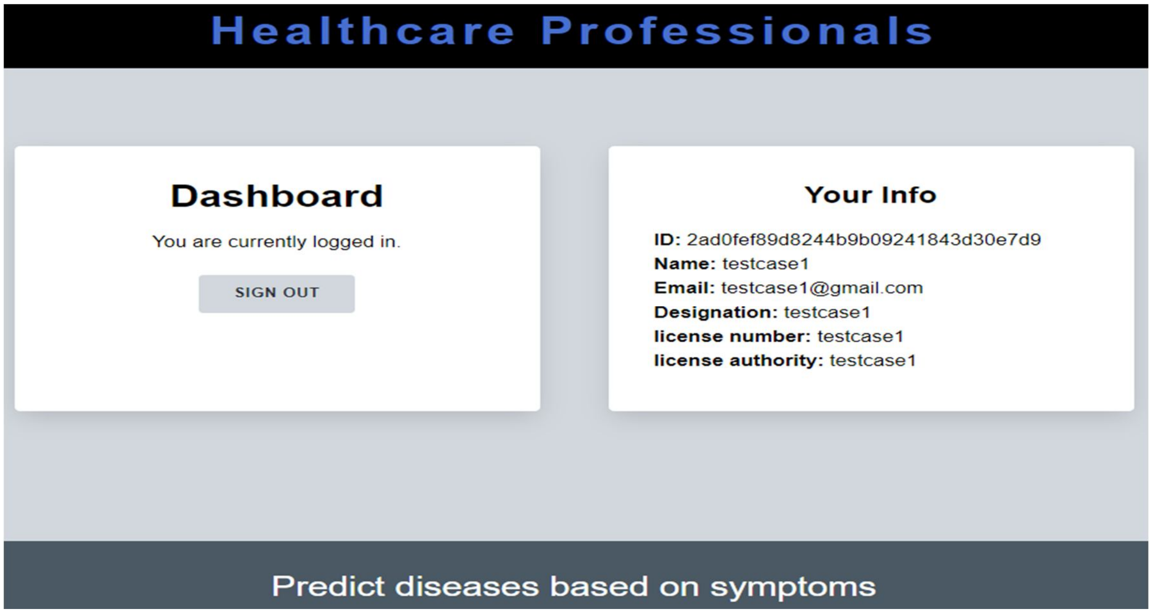
Fig: default values

B. Method of Process

The system follows a structured method to manage Healthcare Professional (HCP) accounts. Initially, a user accesses the HCP Registration form and submits their details including name [5], email, designation, license number, license authority, and password. Once submitted, these details are securely stored in the MongoDB collection, with the password hashed and the approval status set to 0, indicating a pending account. The admin interface then lists all unapproved accounts in the HCP UnApproved LIST, where each entry displays the HCP’s ID and credentials for manual review [11]. The admin can activate an account by entering the HCP’s unique ID into the provided input field and clicking the "ACTIVATE" button. Upon activation, the system updates the approval status to 1, marking the user as verified. After approval, when the HCP logs in, they are redirected to the Dashboard, where their session is confirmed, and personal details are displayed for verification [17]. This method ensures a secure and controlled onboarding process for healthcare professionals in the disease prediction platform

C. Output

After the Healthcare Professional logs in, the system confirms access with the message "You are currently logged in." and displays their registered details, including name, email, designation, and license information. If the account was pending, it initially appears in the Unapproved HCP List with an approval status of 0 [6]. Once the admin activates the account using the HCP ID, the system outputs "account Activated", and the user is granted full access to the dashboard and disease prediction features.



Healthcare Professionals

Dashboard

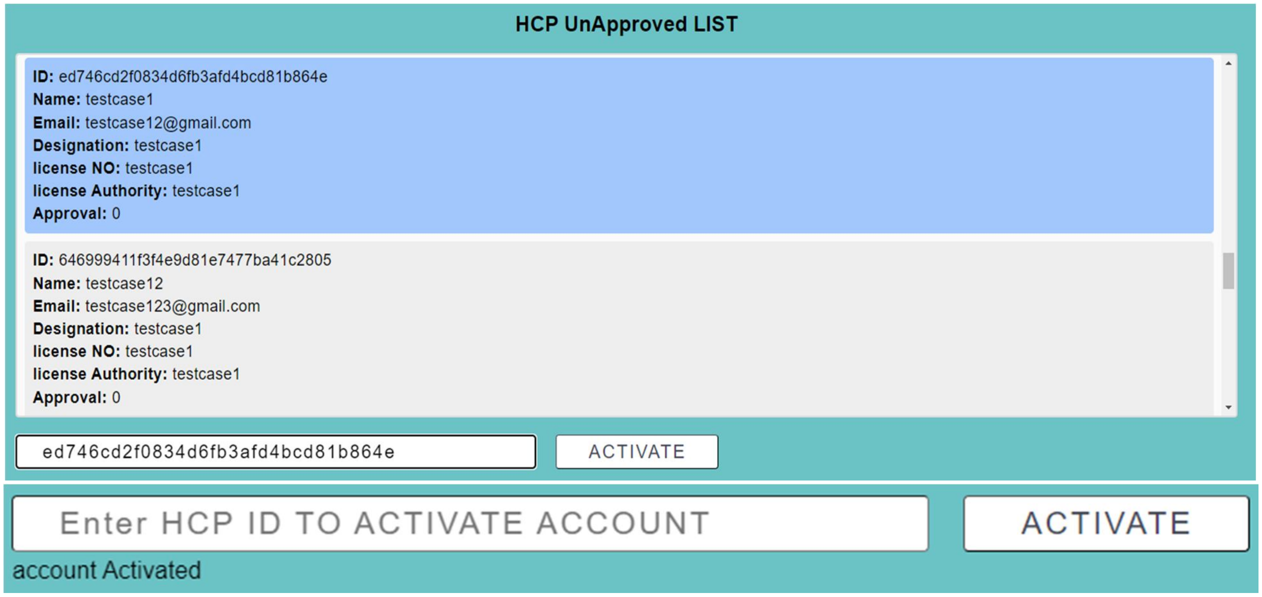
You are currently logged in.

Your Info

ID: 2ad0fef89d8244b9b09241843d30e7d9
Name: testcase1
Email: testcase1@gmail.com
Designation: testcase1
license number: testcase1
license authority: testcase1

Predict diseases based on symptoms

Fig : predict data



HCP UnApproved LIST

ID: ed746cd2f0834d6fb3afd4bcd81b864e
Name: testcase1
Email: testcase12@gmail.com
Designation: testcase1
license NO: testcase1
license Authority: testcase1
Approval: 0

ID: 646999411f3f4e9d81e7477ba41c2805
Name: testcase12
Email: testcase123@gmail.com
Designation: testcase1
license NO: testcase1
license Authority: testcase1
Approval: 0

ed746cd2f0834d6fb3afd4bcd81b864e

Enter HCP ID TO ACTIVATE ACCOUNT

account Activated

IV. RESULTS

After successful registration, the Healthcare Professional can log in and is redirected to the Dashboard, where they are greeted with a confirmation message stating *"You are currently logged in."* On the right panel, the user's personal and professional details such as their Name, Email, Designation, License Number, and License Authority are displayed for reference. Additionally, a unique ID assigned to each HCP is shown to help with account management and verification. For administrative purposes, newly registered HCPs appear in the Unapproved List section. This panel shows all pending users whose accounts are yet to be activated, along with their registration details and approval status (Approval: 0). An admin can approve a user by entering their HCP ID in the provided field and clicking the "Activate" button. Upon successful activation, the system displays a message such as "Account Activated", and the approval status of that HCP is updated in the database.

V. CONCLUSION

The developed Flask application effectively addresses the challenge of disease prediction by providing a user-friendly interface for symptom input and generating potential diagnoses based on a trained machine learning model. By integrating a search functionality and error handling mechanisms, the application enhances user experience and accuracy. The incorporation of detailed disease information, including precautions, medications, and lifestyle recommendations, empowers users to take proactive steps towards their health.

The use of MongoDB Atlas as the database solution ensures efficient data storage and retrieval, supporting user authentication, HCP management, and disease data management. The application's modular structure and clear code organization facilitate maintainability and future enhancements. using above paragraph give me some use full lines

VI. FUTURE SCOPE

While the current application serves as an effective tool for disease prediction, there is significant scope for further enhancement. Future development could focus on improving the model’s accuracy by integrating advanced machine learning techniques such as deep learning and ensemble methods. Expanding the symptom database will ensure broader disease coverage, while personalized recommendations based on user data can promote proactive health management. Additionally, integrating direct consultation features with healthcare providers and developing a mobile application would increase accessibility and user engagement. Incorporating natural language processing (NLP) will enable the system to interpret complex symptom descriptions more accurately.

Lastly, enhancing data privacy and security measures will be essential to maintain user trust and ensure the confidentiality of sensitive health information.

VII. ACKNOWLEDGEMENT

Mr. Rongala Rajesh is an enthusiastic and committed faculty member in the Department of Computer Science. As an early-career academician, he has shown strong dedication to student development through active involvement in project guidance and technical mentoring. Despite being at the beginning of his professional journey, he has effectively guided students in executing academic projects with precision and conceptual clarity. His passion for teaching, coupled with a solid understanding of core computer science principles, positions him as a promising educator and mentor. Mr. Satish continues to contribute meaningfully to the academic environment through his proactive approach to learning and student engagement.

Kuncha Lakshman is pursuing his final semester MCA in Sanketika Vidya Parishad Engineering College, accredited with A grade by NAAC, affiliated by Andhra University and approved by AICTE. With interest in Support Vector classifier K Lakshman has taken up his PG project on SYMPTOMS BASED DISEASE IDENTIFICATION USING SVC and published the paper in connection to the project under the guidance of R Rajesh, Assistant Professor, SVPEC.

REFERENCES

[1] AI-driven healthcare systems: A personalized symptom-based disease prognosis tool using RF, GNB, and SVC techniques https://www.worldscientific.com/doi/full/10.1142/S2737599425500033

[2] Symptoms Based Disease Prediction Using Machine Learning Techniques https://ieeexplore.ieee.org/abstract/document/9388603

[3] A Symptom-based Disease Diagnosis: A Pre-Classification ML Approach https://ieeexplore.ieee.org/abstract/document/10607702

[4] Optimal Symptom Diagnosis for Efficient Disease Identification Using Somars Approach https://www.ingentaconnect.com/contentone/asp/jctn/2017/00000014/00000002/art00040

[5] Support Vectors Machine-based identification of heart valve diseases using heart sounds https://www.sciencedirect.com/science/article/abs/pii/S0169260709000339

[6] The Superior Vena Cava Syndrome Clinical Characteristics and Evolving Etiology https://journals.lww.com/md-journal/fulltext/2006/01000/The_Superior_Vena_Cava_Syndrome__Clinical.00003.aspx

[7] Human symptoms–disease network https://www.nature.com/articles/ncomms5212

[8] Disease Prediction and Early Intervention System Based on Symptom Similarity Analysis https://ieeexplore.ieee.org/abstract/document/8924757

[9] A Symptom-based Disease Diagnosis: A Pre-Classification ML Approach https://ieeexplore.ieee.org/abstract/document/10607702

[10] Initial Evaluation of the Patient With Lung Cancer*: Symptoms, Signs, Laboratory Tests, and Paraneoplastic Syndromes https://www.sciencedirect.com/science/article/abs/pii/S0012369215329871

[11] Rapid detection of fish with SVC symptoms based on machine vision combined with a NAM-YOLO v7 hybrid model https://www.sciencedirect.com/science/article/abs/pii/S004484862400019X

[12] Initial Evaluation of the Patient With Lung Cancer: Symptoms, Signs, Laboratory Tests, and Paraneoplastic Syndromes: ACCP Evidenced-Based Clinical Practice Guidelines (2nd Edition) https://www.sciencedirect.com/science/article/abs/pii/S0012369215355173

[13] Impulse oscillometry indices in relation to respiratory symptoms and spirometry in the Swedish Cardiopulmonary Bioimage Study https://publications.ersnet.org/content/erjor/9/5/00736-2022.abstract

[14] Thromboembolic Disease Involving the Superior Vena Cava and Brachiocephalic Veins https://www.sciencedirect.com/science/article/abs/pii/S0012369216347493

[15] AI-based disease category prediction model using symptoms from low-resource Ethiopian language: Afaan Oromo text https://www.nature.com/articles/s41598-024-62278-7

[16] Incidence and predictive factors of symptomatic thrombosis related to peripherally inserted central catheters in chemotherapy patients https://www.sciencedirect.com/science/article/abs/pii/S0049384812001168

[17] GAINSeq: glaucoma pre-symptomatic detection using machine learning models driven by next-generation sequencing data https://www.nature.com/articles/s41598-025-04249-0

[18] Validation of the Symptom Pattern Method for Analyzing Verbal Autopsy Data https://journals.plos.org/plosmedicine/article?id=10.1371/journal.pmed.0040327

[19] Pediatric superior vena cava syndrome: An evidence-based systematic review of the literature https://onlinelibrary.wiley.com/doi/abs/10.1002/pbc.27225

[20] Coronary Heart Disease Prediction Using Voting Classifier Ensemble Learning https://ieeexplore.ieee.org/abstract/document/9725705

[21] Complications of Vascular Access: Superior Vena Cava Syndrome https://www.sciencedirect.com/science/article/abs/pii/S0272638616305613