iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** XI **Month of publication:** November 2026

DOI: https://doi.org/10.22214/ijraset.2026.85608

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue XI Nov 2026- Available at www.ijraset.com

AI-Assisted Quantum-Enhanced 5G Smart Healthcare System for Multi-Symptom Disease Prediction

B. Haseena Begam[1], Md. A. Khudhus[2], Dr. M. Humera Khanam[3], Dr. D. Vivekanandha Reddy[4]
[1, 2, 3, 4]*Department of Computer Science and Engineering, Sri Venkateswara University College of Engineering, Tirupati, A.P*

Abstract --The growing need for accessible and efficient healthcare services has driven the creation of intelligent systems that can predict diseases in their early stages. This paper introduces an AI-enhanced smart healthcare system designed to predict multiple diseases based on symptoms reported by users. The system is built using React.js and JavaScript, and it relies on a structured JSON dataset that includes disease names, symptoms, causes, medicines, and risk levels. It matches user-provided symptoms with existing disease records, computes symptom similarity scores, and shows the most likely disease along with related health details. The system also includes user authentication and a record of previous predictions to ensure ease of access and better data management. The evaluation dashboard indicates that the system achieved a validation accuracy of 96.82%, macro-precision of 96.48%, macro-recall of 95.69%, and a macro F1-score of 95.54% over 12,060 test cases. This represents a significant improvement from the previous project's validation accuracy of 80.26%, showing an increase of 16.56 percentage points. The study also outlines the potential of 5G communication for enhancing connected healthcare services and Grover's quantum search algorithm for future optimization in search processes. However, these technologies are currently conceptual and the system still relies on conventional JavaScript-based symptom matching. The objective of the system is to provide users with a user-friendly interface for accessing initial health information. However, it is explicitly stated that its predictions should not be used as a substitute for professional medical diagnosis or treatment.

Keywords: Artificial Intelligence, Smart Healthcare, Multi-Symptom Disease Prediction, Symptom Matching, Classification Accuracy, Precision, Recall, F1-Score, 5G Communication, Quantum Computing.

I. INTRODUCTION

Digital technologies have opened up new ways to make it easier to get to and understand health information. Identifying a possible disease from multiple symptoms can be challenging because different conditions may produce similar symptoms. Artificial Intelligence (AI) can support this process by analyzing symptom information and identifying patterns that may help users understand possible health conditions.

An AI-assisted smart healthcare system for multi-symptom disease prediction is presented in this paper. React.js and JavaScript are used in the application's development, and a structured JSON dataset with disease names, symptoms, causes, medication details, and risk levels is used. It displays the closest matching result with a symptom-match score after comparing user-entered symptoms with stored records. The system also provides user authentication and prediction history.

The evaluation dashboard reports 96.82% validation accuracy, 96.48% macro-precision, 95.69% macro-recall, and a 95.54% macro F1-score across 12,060 test cases. The project also explores the possible future use of 5G communication and Grover's quantum search algorithm. These are still conceptual extensions; the current prototype uses traditional JavaScript-based symptom matching.

The proposed system aims to provide convenient access to preliminary health information through a simple web interface. Its predictions are intended to provide information and should not be used as a substitute for professional medical advice or diagnosis.

II. RELATED WORK

1) *Artificial Intelligence in Healthcare:* Artificial intelligence has become an important research area in healthcare because it can analyze medical data, identify patterns, and support preliminary disease prediction. AI-based systems can help organize health information and assist users in understanding possible medical conditions. However, their performance is dependent on the dataset's quality, prediction method, and proper validation. The World Health Organization also emphasizes safety, transparency, and responsible use of AI in healthcare [6], [7].

©IJRASET: All Rights are Reserved

2) *Symptom-Based Disease Prediction:* Digital symptom assessment systems use the symptoms reported by users to suggest potential health conditions. These systems can improve access to preliminary health information and help users understand when additional medical evaluation is required. However, different diseases can exhibit similar symptoms, leading to inaccurate predictions. Research on digital symptom assessment emphasizes the importance of assessing prediction accuracy and the reliability of the provided information [8].
3) *5G Communication in Healthcare:* Fifth-generation (5G) communication technology has the potential to improve connectivity between healthcare professionals, patients, and connected medical devices. Its high data-transfer capacity and low-latency capabilities can support applications such as remote patient monitoring, telemedicine, and real-time health information exchange. Research has discussed both the opportunities and challenges of adopting 5G in healthcare [4]. In the proposed system, 5G communication is considered a potential enhancement rather than an implemented feature.
4) *Quantum Search Techniques:* Under suitable assumptions, Grover's quantum search algorithm was presented as a way to search an unstructured search space with fewer oracle queries than classical exhaustive search [2]. This has prompted investigations into potential uses of quantum computing for optimization and search issues. However, applying Grover's algorithm to healthcare datasets requires suitable quantum-computing infrastructure, implementation, and experimental comparison. In the current project, quantum search is discussed as a future research direction and is not part of the implemented prediction process.
5) *Research Gap:* Despite earlier research on AI-based healthcare applications, digital symptom assessment, 5G communication, and quantum search techniques, challenges remain in integrating accessible symptom input, disease information, risk levels, and prediction history into a single user-friendly application. Prediction reliability, dataset completeness, and independent evaluation also require further attention. The proposed system addresses these needs by using React.js, JavaScript, and a structured JSON dataset to compare multiple user-entered symptoms with available disease records and display the closest matching result. It also provides user authentication and prediction history. However, rather than using a trained machine-learning model, the current prototype relies on conventional symptom-matching rules.Further research should focus on clinically verified datasets, independent performance evaluation, stronger data protection, and the practical implementation of advanced healthcare communication and search technologies.

III. PROBLEM DEFINITION AND OBJECTIVES

Accessing pertinent health information can be time-consuming, and different diseases may share similar symptoms, making it difficult to identify potential diseases from multiple symptoms. It may also be challenging to compare multiple disease records and identify the closest matching condition when using manual symptom assessment. Through a single platform, users may encounter challenges in comprehending potential causes, risk levels, and associated medical information. The project's objective is to create a smart healthcare application that is AI-assisted and simplifies the preliminary disease prediction process by comparing user-entered symptoms with a structured disease dataset.

To develop a user-friendly web application for multi-symptom disease prediction, the following are the objectives:

1) Provide a structured disease dataset that includes symptoms, disease names, causes, medicine information, and risk levels.
2) Process user-entered symptoms and identify the closest matching disease record.
3) Display prediction results with relevant disease information and symptom-match scores.
4) Implement user registration and login for application access.
5) Maintain prediction history for reviewing previous results.
6) Evaluate system performance using accuracy, precision, recall, and F1-score.
7) Enhance prediction reliability through clinically verified datasets and advanced machine-learning techniques.
8) Explore the potential future integration of 5G communication and Grover's quantum search algorithm for connected healthcare and suitable search applications.

IV. PROPOSED SYSTEM

The smart healthcare web application that is being proposed is an AI-assisted system that is intended to offer users preliminary disease predictions based on a variety of symptoms that they will enter. It is created using JavaScript, React.js, and a structured JSON dataset that includes disease names, associated symptoms, potential causes, medicine information, and risk levels. The closest matching result, along with relevant health information and a symptom-match score, is displayed by the system after comparing user-entered symptoms with the available disease records.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Additionally, it offers user registration, login, and prediction history to enhance convenience and accessibility. Metrics such as precision, recall, accuracy, and F1-score can be employed to evaluate system performance. The current prototype employs conventional JavaScript-based symptom matching, while Grover's quantum search algorithm, real-time 5G communication, and trained machine-learning models are considered future enhancements. The system's objective is to increase the accessibility of preliminary health information, while also emphasizing that its predictions do not serve as a substitute for professional medical diagnosis or treatment. Figure 1 illustrates the architecture of the 5G Smart Healthcare system.

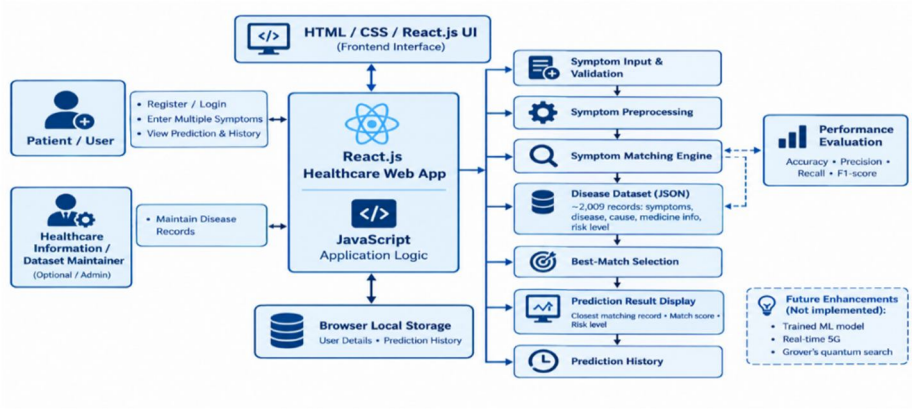


Fig. 1. 5G Smart Healthcare system architecture

A. *User Authentication and Dashboard Module*

This module is responsible for the registration, login, and access of users to the healthcare application. The dashboard, which offers navigation to symptom input, disease prediction, prediction results, and history, can be accessed by users by entering their credentials. The interface is developed using React.js, while user interactions and form validation are managed by JavaScript. Basic user information management is accomplished through the utilization of browser Local Storage in the current prototype. The dashboard is intended to offer users a user-friendly and straightforward experience.

B. *Symptom Input and Processing Module*

This module enables users to input a variety of symptoms that are associated with their health condition. The symptoms are collected and processed by the application prior to the disease matching process. By converting text to a consistent case and removing unnecessary spaces, JavaScript can standardize symptom names. Before submitting a prediction request, input validation checks that users have provided symptoms. The input is prepared for comparison with the disease dataset, and mismatches caused by differences in text formatting are reduced through proper symptom processing.

C. *Disease Prediction and Matching Module*

The core module of the proposed system is this. It compares the symptoms entered by the user with the symptom information contained in the structured JSON dataset. JavaScript-based matching rules determine scores by comparing the symptoms entered with the symptoms associated with each disease record. The system subsequently selects the record that is closest to the selected record and prepares it for display. An appropriate message can be displayed if no suitable match is found. The matching score should not be interpreted as a medically validated probability of disease, as the current prototype employs conventional symptom-matching techniques rather than a trained machine-learning model.

D. *Prediction Results and History Module*

The output produced by the disease-matching process is displayed in this module. The outcome comprises the disease that is most closely comparable to the input and the pertinent information that is available, including potential causes, medication details, risk levels, and symptom-match scores. In an effort to facilitate comprehension of the outcome, the interface presents these particulars in a manner that is easily comprehensible. Prediction history is also maintained by the module, enabling users to review previous results. The current prototype retains pertinent information through the utilization of Browser Local Storage. Before handling sensitive medical records in a real healthcare environment, it would be necessary to implement more robust security measures, as Local Storage offers limited security.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

E. Performance Evaluation Module

This module facilitates the evaluation of the system by employing performance metrics, including precision, recall, accuracy, and F1-score. Precision measures the accuracy of predicted classes, recall measures the number of actual cases identified, and the F1-score combines precision and recall. The evaluation dashboard for the project indicates that the macro F1-score is 95.54%, the validation accuracy is 96.82%, the macro-precision is 96.48%, and the macro-recall is 95.69%, based on 12,060 test cases. These figures should be interpreted in accordance with the test dataset and evaluation methodology, which should be confirmed for reproducibility. Before drawing conclusions regarding clinical reliability, additional testing with independent and clinically verified datasets is necessary.

V. SYSTEM ARCHITECTURE AND FUNCTIONAL WORKFLOW

The system architecture describes the overall structure of the AI-Assisted Smart Healthcare System for Multi-Symptom Disease Prediction and the interaction between its components. The application is developed using React.js, JavaScript, CSS, and a structured JSON disease dataset. React.js provides the user interface, while JavaScript manages user interactions, symptom processing, disease matching, and result generation.

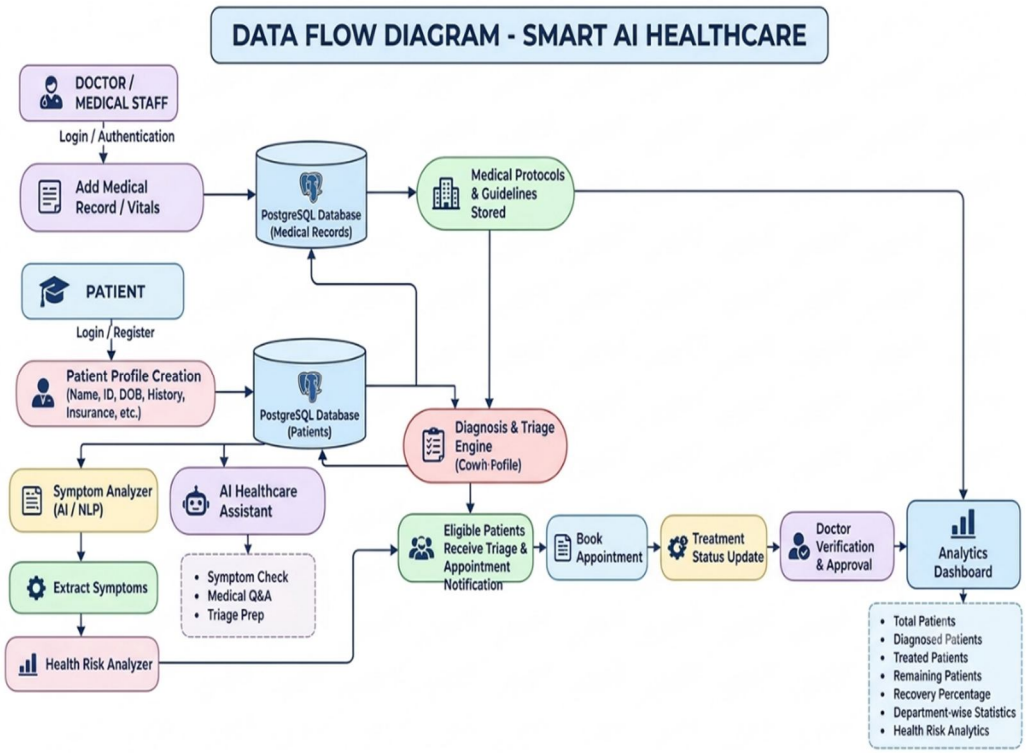


Fig. 2. Data flow of 5G Smart Healthcare System.

The architecture consists of five main components: the user interface, symptom processing, disease prediction, data storage, and performance evaluation. Users register or log in and enter multiple symptoms through the application interface. The symptom processing component validates and standardizes the input before passing it to the disease-matching component. The matching component compares the symptoms against records in the JSON dataset and identifies the closest matching disease. The result is then displayed with available disease information, causes, medicine details, risk levels, and a symptom-match score. Browser Local Storage maintains basic user information and prediction history in the current prototype. The performance evaluation component presents metrics such as accuracy, precision, recall, and F1-score according to the configured testing procedure. Real-time 5G communication, trained machine-learning models, and Grover's quantum search algorithm are considered future enhancements rather than implemented components.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

A. Technology Stack

Technology	Role	Implementation Use
React.js	Frontend Framework	Development of the patient dashboard, symptom input interface, and prediction result pages
HTML / CSS	User Interface	Structuring web pages, styling components, and designing a responsive healthcare interface
JavaScript	Application Logic	Processing user input, matching symptoms, calculating match scores, and displaying results
JSON	Dataset	Storing disease names, symptoms, causes, medicine information, and risk levels
Browser Local Storage	Data Storage	Maintaining basic user information and prediction history
Node.js / npm	Development Tools	Running the React development environment and managing project dependencies
Accuracy, Precision, Recall, F1-Score	Performance Evaluation	Assessing and presenting system performance using the configured test data
5G Communication	Future Technology	Potential support for connected healthcare and remote health information exchange
Grover's Algorithm	Future Technology	Exploring quantum search techniques for suitable search applications

VI. SYSTEM IMPLEMENTATION

System implementation is the process of converting the proposed system design into a functional software application. The AI-Assisted Smart Healthcare System for Multi-Symptom Disease Prediction is implemented as a web-based application that helps users obtain preliminary health information by entering their symptoms. The system is developed using React.js, JavaScript, CSS, and a structured JSON dataset. React.js is used to build the user interface and manage the different application screens, JavaScript implements the application logic, CSS provides the visual design, and the JSON dataset stores disease-related information.

A. User Registration and Login Module

This module allows users to register and log in to the healthcare application. User details are validated, and successful login provides access to the patient dashboard. React.js manages the interface, while JavaScript handles form inputs and authentication logic. Browser Local Storage is used in the current prototype to store user information.

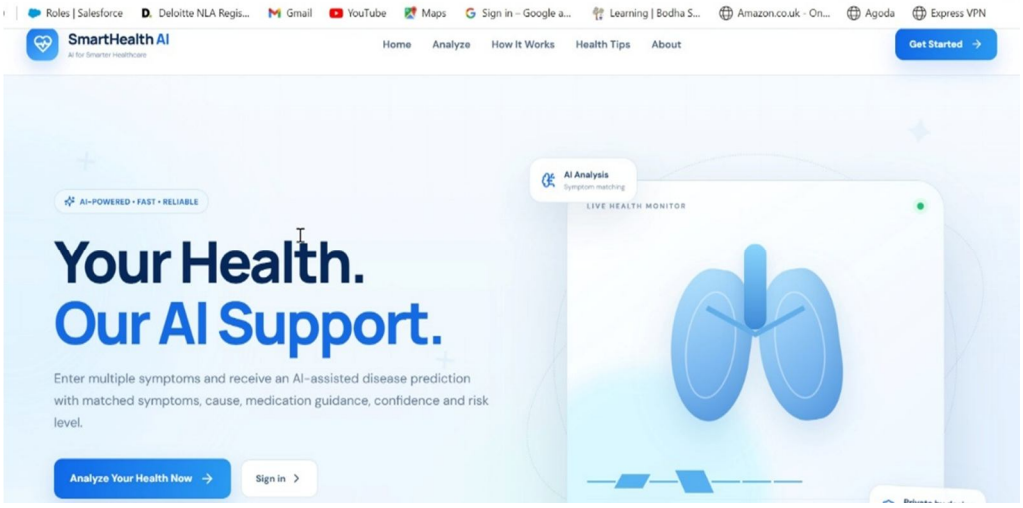


Fig.1.Home page

B. Patient Dashboard Module

The dashboard provides access to the main application features, including symptom entry, disease prediction, results, and prediction history. React.js components and CSS are used to create a simple, responsive interface that allows users to navigate between different sections easily.

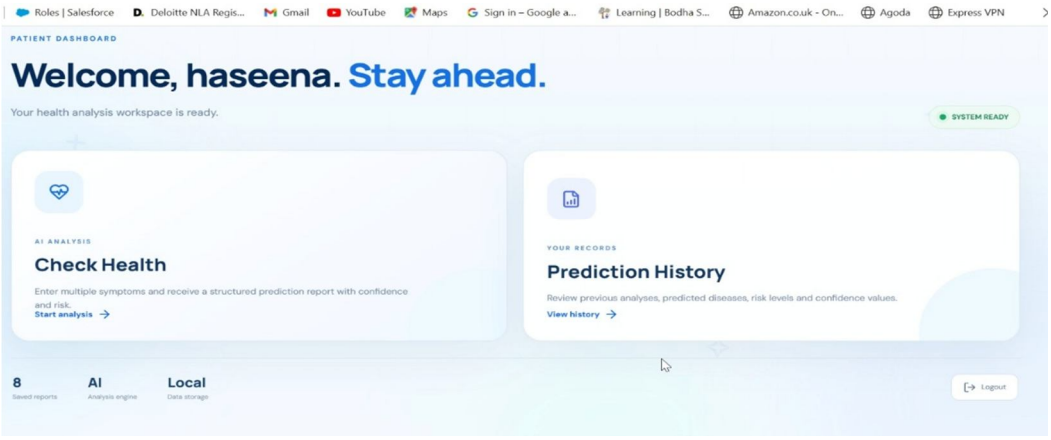


Fig.2.Dashboard Page

C. Symptom Input Module

This module collects multiple symptoms entered by the user. JavaScript processes the input by standardizing symptom names and preparing them for comparison with the disease dataset. Input validation helps prevent empty or incomplete submissions.

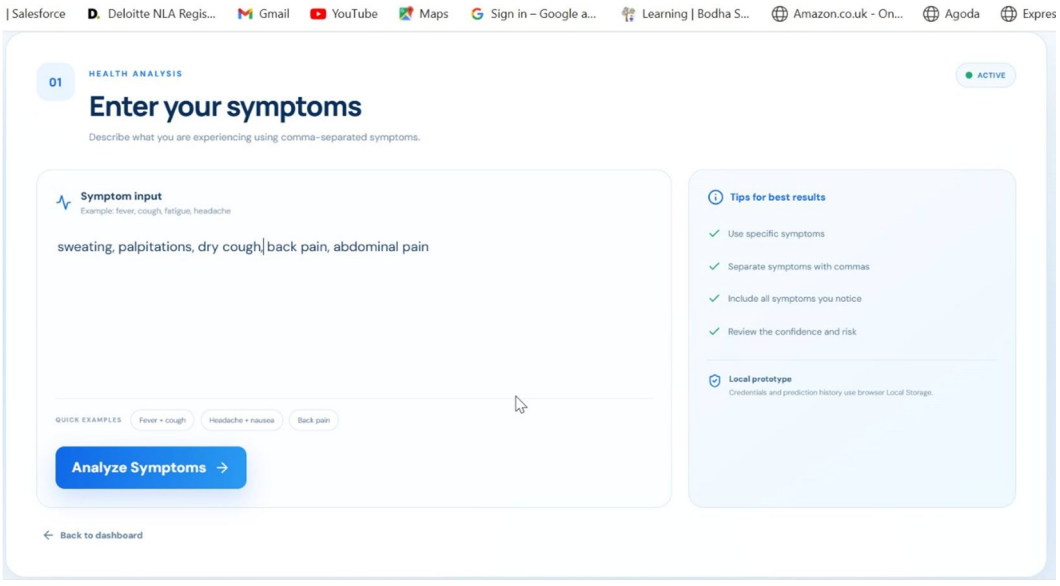


Fig.3.Symptoms Page

D. Disease Prediction Module

This is the main processing module of the system. It compares user-entered symptoms with the symptoms stored in the dataset and calculates matching scores according to the implemented rules. The closest matching disease record is selected and used to generate the prediction result. The current prototype uses conventional JavaScript-based symptom matching rather than a trained machine-learning model.

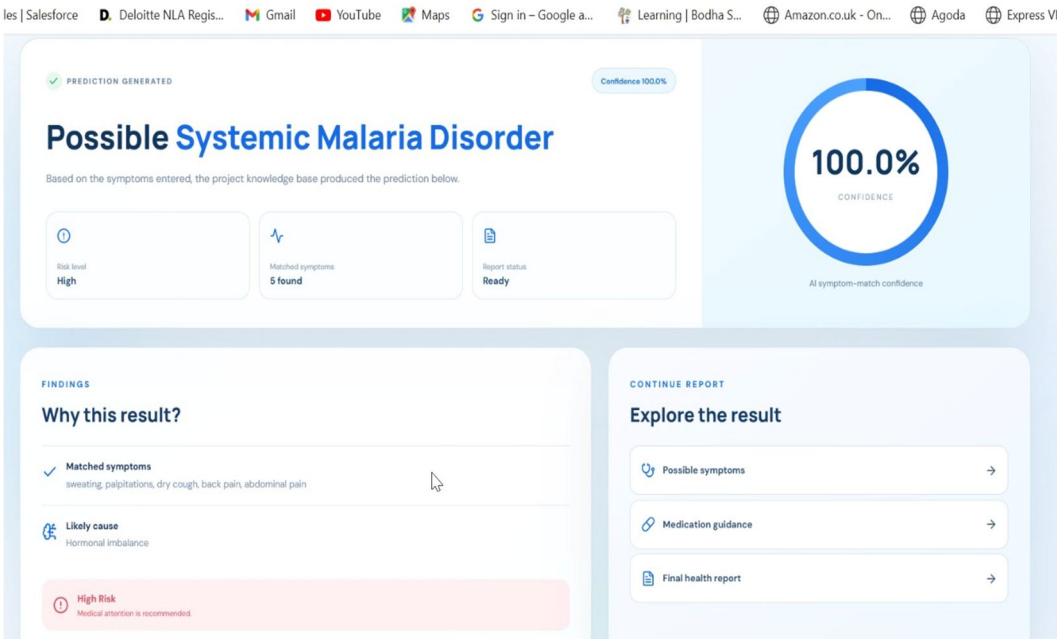


Fig.4. Disease Prediction

E. Prediction Results Module

This module displays the selected disease record and its available details, including symptoms, possible causes, medicine information, risk level, and symptom-match score. The results are presented in a readable format to help users understand the information. These predictions provide preliminary health information and do not replace professional medical diagnosis.

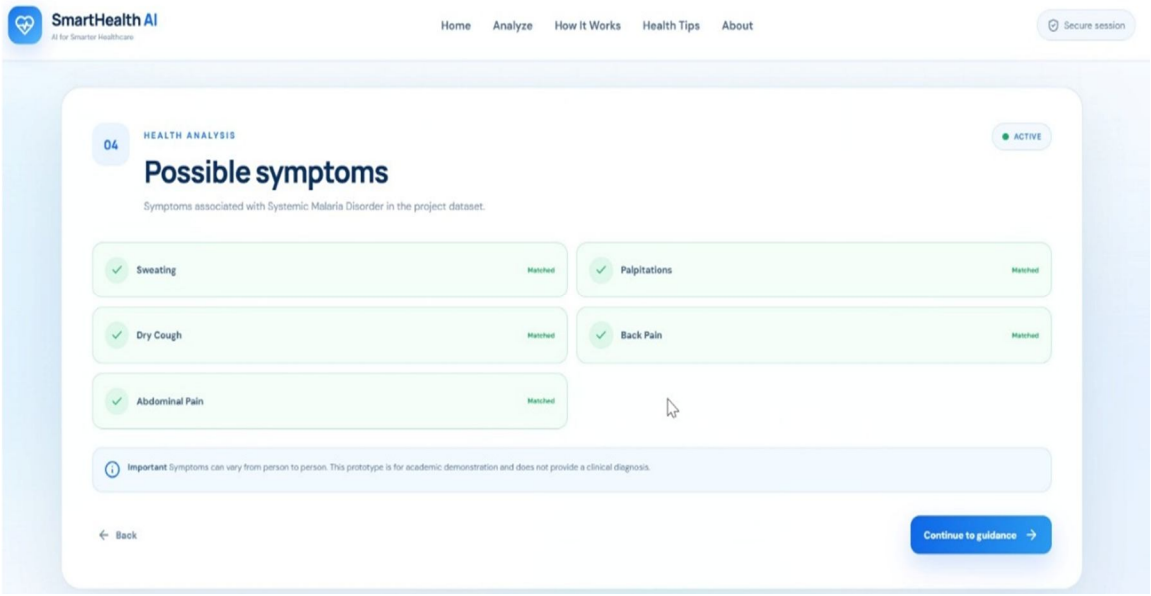


Fig.5.Possible Symptoms

A. Prediction History Module

This module enables users to view previous prediction records. Browser Local Storage is used to retain relevant history data, allowing the application to retrieve and display earlier results. Stronger security measures would be required before storing sensitive medical information in a production healthcare system.

©IJRASET: All Rights are Reserved

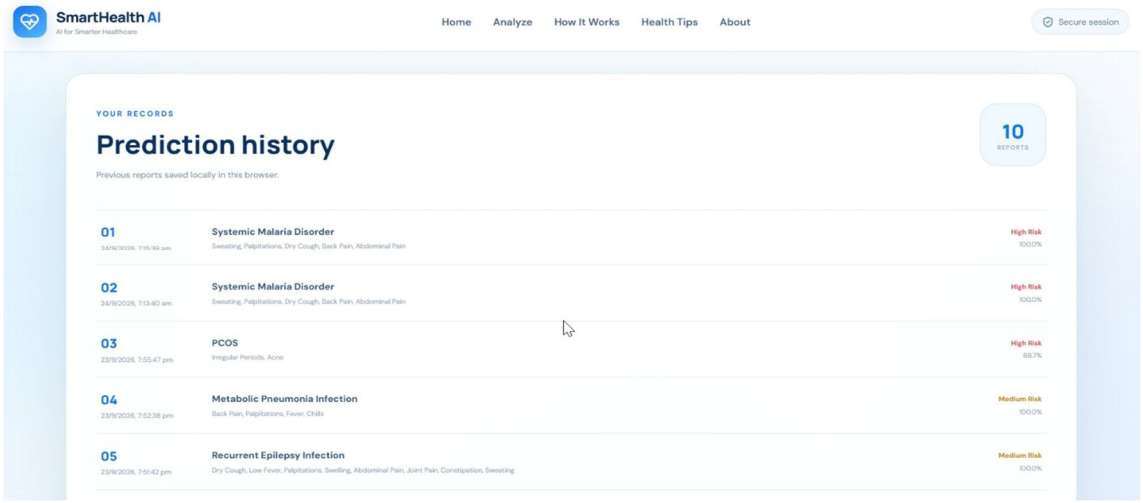


Fig.6. Prediction History

VII. RESULTS AND DISCUSSION

The proposed AI-assisted smart healthcare system was developed using React.js, JavaScript, and a structured JSON dataset containing disease names, associated symptoms, causes, medicine information, and risk levels. The application allows users to enter multiple symptoms and compares them with the available disease records to identify the closest matching condition. The results are displayed through a user-friendly interface, providing relevant disease information and a symptom-match score. The system also includes user authentication and prediction history to improve accessibility and support the review of previous predictions. According to the evaluation dashboard, the updated system achieved a validation accuracy of 96.82%, macro-precision of 96.48%, macro-recall of 95.69%, and a macro F1-score of 95.54% across 12,060 test cases. The dashboard also reports that accuracy increased from 80.26% in the earlier version to 96.82% in the updated version, representing an improvement of 16.56 percentage points. These reported results suggest improved performance under the evaluation conditions used. However, the metrics should be interpreted according to the dataset, test procedure, and evaluation method employed, and independent validation is required to confirm generalizability. The results demonstrate the potential of a dataset-based symptom-matching application to provide convenient preliminary health information. However, the current prototype uses conventional JavaScript-based matching rather than a fully trained machine-learning model. Similarly, 5G communication and Grover's quantum search algorithm are considered potential future enhancements and are not implemented in the present system. Overall, the application provides a foundation for further development of accessible healthcare support tools, but its predictions should not be treated as confirmed diagnoses or replacements for professional medical advice.

VIII. SYSTEM TESTING

A. Unit Testing

Unit testing focused on authentication, OTP validation, eligibility checking, recommendation handling, and student skill updates. Individual components were tested independently before evaluating their interaction within the complete application.

B. Functional Testing

Functional testing examined registration, login, profile management, placement-drive access, eligibility verification, career recommendations, skill-gap features, notifications, and mentor analytics. The objective was to verify that the major application operations behaved according to the defined requirements.

C. Integration Testing

Integration testing verified communication between the frontend, Django backend, PostgreSQL database, and AI services. It included registration and authentication, eligibility calculations, communication with Ollama, fallback recommendation behaviour, and mentor dashboard polling.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

D. Performance Testing

Performance testing considered database access, eligibility calculations, AI response generation, recommendation retrieval, and dashboard updates. The locally hosted language model produced responses in approximately two to six seconds under the documented test conditions. Recommendation caching reduces repeated AI work when the student profile remains unchanged.

Test Type	Test Case
Unit Testing	Test individual functions such as login validation and symptom matching.
Integration Testing	Check interaction between React.js, JavaScript, the JSON dataset, and Local Storage.
Functional Testing	Test registration, login, symptom entry, prediction, and history features.
Input Validation Testing	Submit empty, incomplete, or invalid symptom inputs.
Disease Prediction Testing	Enter multiple symptoms and compare them with dataset records.
Result Testing	Verify displayed disease information, causes, medicine details, risk levels, and match scores.
History Testing	Generate predictions and retrieve previously saved results.
Performance Evaluation	Assess accuracy, precision, recall, and F1-score using the configured test procedure.
Usability Testing	Check navigation, readability, and ease of entering symptoms.
Security Testing	Check credential handling, input validation, and stored user information.
Validation Testing	Compare predictions against reliable reference labels using suitable test data.

IX. LIMITATIONS

The proposed AI-assisted smart healthcare system has certain limitations. Its prediction results depend on the quality, completeness, and accuracy of the available disease dataset. Since the current prototype uses conventional symptom-matching techniques, it may not accurately identify complex diseases or distinguish between conditions that share similar symptoms. The medical information, including causes, medicine details, and risk levels, is also limited to the records available in the dataset. The system does not fully consider individual patient factors such as medical history, age, allergies, or laboratory test results. Furthermore, the predictions have not been established as a substitute for professional medical diagnosis and require further clinical validation. User information and prediction history stored in browser Local Storage have limited security and need stronger protection for handling sensitive health data. The current prototype does not implement real-time 5G communication or Grover's quantum search algorithm; these are considered potential future enhancements. Although the evaluation dashboard reports promising performance metrics, further testing with independent and clinically verified datasets is necessary to establish the system's reliability. Therefore, the application should be used for preliminary health information and awareness rather than as a replacement for professional medical advice or treatment.

X. FUTURE ENHANCEMENT

The proposed system can be enhanced by integrating trained machine-learning models to improve multi-symptom disease prediction and validating their performance using diverse, clinically verified datasets. Future versions may incorporate real-time 5G communication for remote healthcare connectivity and explore quantum search techniques, including Grover's Algorithm, through suitable quantum-computing platforms. Additional features could include wearable-device integration, multilingual support, secure cloud-based health records, improved privacy protection, and connections with healthcare professionals for further assessment. The prediction interface could also be enhanced with personalized health guidance and improved risk assessment. These developments could make the system more reliable, accessible, and useful while ensuring that all predictions remain subject to appropriate medical validation.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

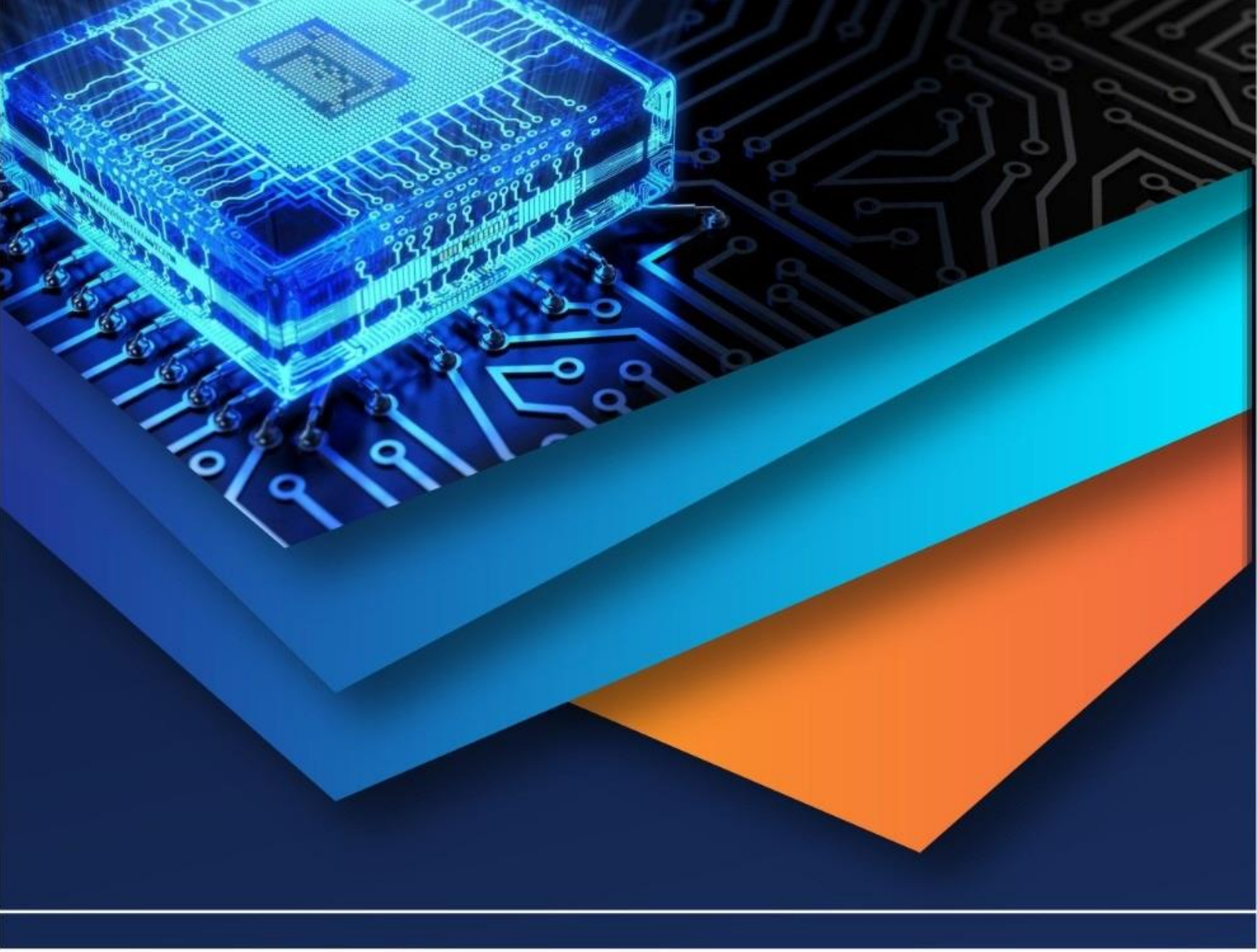
XI. CONCLUSION

This paper presents an AI-assisted smart healthcare system for multi-symptom disease prediction using React.js, JavaScript, and a structured JSON dataset. The application compares user-entered symptoms with stored disease records and displays the closest matching disease, associated information, risk level, and match score. The evaluation dashboard reports 96.82% validation accuracy, 96.48% macro-precision, 95.69% macro-recall, and a 95.54% macro F1-score across 12,060 test cases. These results indicate promising performance under the reported evaluation conditions. The system also provides user authentication and prediction history for convenient access. Although 5G communication and Grover's quantum search algorithm are considered potential future enhancements, they are not implemented in the current prototype. Further work will focus on independent validation, improved prediction methods, and stronger data privacy. The system is intended to support preliminary health awareness and does not replace professional medical diagnosis or treatment.

REFERENCES

[1] J. McCarthy, "What is artificial intelligence?" Stanford University, 2007.
[2] L. K. Grover, "A fast quantum mechanical algorithm for database search," in Proceedings of the 28th Annual ACM Symposium on Theory of Computing, 1996, pp. 212–219.
[3] "An AI-assisted smart healthcare system using 5G communication," IEEE Access, vol. 11, 2023.
[4] A. Moglia, K. Georgiou, B. Marinov, E. Georgiou, R. N. Berchiolli, R. M. Satava, and A. Cuschieri, "5G in healthcare: From COVID-19 to future challenges," IEEE Journal of Biomedical and Health Informatics, vol. 26, no. 8, pp. 4187–4196, 2022.
[5] N. Taimoor and S. Rehman, "Reliable and resilient AI and IoT-based personalised healthcare services: A survey," IEEE Access, vol. 10, 2022.
[6] World Health Organization, Ethics and Governance of Artificial Intelligence for Health: WHO Guidance. Geneva, Switzerland: WHO, 2021.
[7] World Health Organization, Regulatory Considerations on Artificial Intelligence for Health. Geneva, Switzerland: WHO, 2023.
[8] S. Gilbert et al., "How accurate are digital symptom assessment apps for suggesting conditions and urgency advice? A clinical vignettes comparison to GPs," BMJ Open, vol. 10, no. 12, Art. no. e040269, 2020.
[9] M. A. Gianfrancesco et al., "Potential biases in machine learning algorithms using electronic health record data," JAMA Internal Medicine, vol. 178, no. 11, pp. 1544–1547, 2018.
[10] B. A. Goldstein et al., "Opportunities and challenges in developing risk prediction models with electronic health records data: A systematic review," Journal of the American Medical Informatics Association, vol. 24, no. 1, pp. 198–208, 2017.
[11] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.
[12] C. M. Bishop, Pattern Recognition and Machine Learning. New York, NY, USA: Springer, 2006.
[13] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. New York, NY, USA: Springer, 2009.
[14] S. Raschka, Y. Liu, and V. Mirjalili, Machine Learning with PyTorch and Scikit-Learn. Birmingham, U.K.: Packt Publishing, 2022.
[15] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.
[16] M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information, 10th anniversary ed. Cambridge, U.K.: Cambridge University Press, 2010.
[17] A. Montanaro, "Quantum algorithms: An overview," npj Quantum Information, vol. 2, Art. no. 15023, 2016.
[18] 3GPP, NR; Overall Description; Stage-2, Technical Specification 38.300, 3rd Generation Partnership Project.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)