Journal of Informatics and Web Engineering

Vol. 3 No. 1 (February 2024) eISSN: 2821-370X

Personalized Healthcare: A Comprehensive Approach for Symptom Diagnosis and Hospital Recommendations Using AI and Location Services

Seng-Keong Tan[1], Siew-Chin Chong[1*], Kuok-Kwee Wee[1], Lee-Ying Chong[1]

[1]Faculty of Information Science & Technology, Multimedia University, Malaysia

**Corresponding author: (chong.siew.chin@mmu.edu.my, ORCiD: 0000-0003-0421-4367)*

Abstract - Utilizing digital advancements, an integrated Flask-based platform has been engineered to centralize personal health records and facilitate informed healthcare decisions. The platform utilizes a Random Forest model-based symptom checker and an OpenAI API-powered chatbot for preliminary disease diagnosis and integrates Google Maps API to recommend proximal hospitals based on user location. Additionally, it contains a comprehensive user profile encompassing general information, medical history, and allergies. The system includes a medicine reminder feature for medication adherence. This innovative amalgamation of technology and healthcare fosters a user-centric approach to personal health management.

Keywords— Random Forest model, OpenAI API, personal health records, symptom diagnosis, Google maps API

Received: 14 June 2023; Accepted: 19 September 2023; Published: 16 February 2024

I. INTRODUCTION

Malaysia offers almost free healthcare to all of its citizens. The system is not, however, based on a federal insurance program. Instead, the cost of treatment in public facilities is highly subsidized by the government. A well-established commercial system and a publicly financed healthcare system can complement one another well.

The digital era has begun to influence the healthcare industry heavily. The creation of healthcare applications has quickly become a vital component of the sector. Although it streamlines medical procedures, it also has the potential to greatly enhance patient satisfaction. The creation of healthcare application has a wide range of benefits. Access to health data is increased, as is remote health delivery, the reduction of medical errors, cost savings, greater communication, and the list goes on. The creation of healthcare application is now essential for boosting patient happiness and revenue.

Journal of Informatics and Web Engineering

https://doi.org/10.33093/jiwe.2024.3.1.8

Published by MMU Press. URL: https://journals.mmupress.com/jiwe

The digital transformation of healthcare systems is a good trend that is quickly gaining traction among many hospitals, healthcare organizations, and institutions. The overarching goal is to provide remote, round-the-clock access to medical services. Reduce expense in healthcare, enhancing physician-patient communication, track and keep an eye on health conditions, reduce medical mistake, increase accessibility to everyone.

The market is being shaped and expanded by these quickly developing segments of the digital healthcare sector [1]. Healthcare investors have a fantastic chance to advance in any feasible industry with the rise of Artificial Intelligence and Machine Learning. Sophisticated web-based solutions can be developed to enhance decision-making, improve medical accuracy, and enable personalization. This helps therapists correctly identify patients and select the best course of action. The system can provide specific concerns based on specified symptoms that can help doctors reach a decision. Medicine reminder also may begin to play a role in life of all not just the elderly but also adult that is busy with work as life expectancy getting higher.

Myriad of healthcare services are being integrated with technology, such as Electronic Health Record replace paperwork medical record, algorithm, Artificial Intelligence, medical devices, Cloud technology etc. This paper will focus on Symptom checker with Machine Learning, Symptom Checker with OpenAI API and Google Maps API, personal health record, medicine reminder. In the context of the topic "Personalized Healthcare: A Comprehensive Approach for Symptom Diagnosis and Hospital Recommendations using AI and Location Services," the term "comprehensive" refers to an all-encompassing or thorough strategy that covers various aspects of healthcare. It implies that the approach takes into account a wide range of factors, such as symptoms, AI-driven diagnosis, location-based services, and hospital recommendations, in order to provide a holistic and thorough solution to healthcare needs. In essence, it suggests that the approach leaves no important aspects or considerations unaddressed when delivering personalized healthcare services.

The remainder of this paper is structured as follows. Literature works are discussed in Section 2. Section 3 presents the methodology of the proposed system while Section 4 elaborates the overview and design of the proposed system. Section 5 demonstrates the system previews and description. Finally, conclusion is given in Section 6.

II. LITERATURE REVIEW

A. Personal Health Record

In Year 2022, the work [2] aimed to assist kidney transplant patients (KTPs) in managing their chronic conditions and self-care challenges using personal health records (PHRs). The applied-developmental research consisted of three phases. In the first phase, the researchers designed an initial KTP health record (KTPHR) model based on relevant resources and expert opinions. In the second phase, they developed a KTPHR app for Android, which featured health records, a data dashboard, test results, medication tracking, appointment scheduling, and training. The third phase evaluated the usability of the KTPHR using think-aloud and heuristics techniques. The study found qualitative evaluation to be more reliable than quantitative heuristics, with "flexibility and efficiency of use" ranking highest in severity. The final KTPHR model, validated by clinical and basic science specialists, aimed to improve self-care behaviors in KTPs and accelerate clinician decision-making. The researchers concluded that the model met the main evaluation criteria, such as content validity and usability, making it a confident and reliable tool.

B. Online Symptom Checker

A novel approach has been proposed by [3] to disease detection, specifically for COVID-19, using machine learning algorithms to analyse symptoms, thereby circumventing the need for laboratory tests. They applied six supervised machine learning algorithms namely decision tree, random forest, support vector machine, k-nearest neighbours, naive Bayes algorithms, and artificial neural networks to the "COVID-19 Symptoms and Presence Dataset" from Kaggle. Utilizing hyperparameter optimization and 10-fold cross-validation, they compared the algorithms based on accuracy, sensitivity, specificity, and area under the ROC curve. The researchers found that random forest, support vector machine, k-nearest neighbours, and artificial neural networks outperformed the other algorithms, achieving up to 98.84% accuracy. A web application was then developed, allowing users to input their symptoms and receive a real-time prediction on the presence of COVID-19 based on the developed prediction model. The system effectively demonstrates the potential of machine learning algorithms in disease detection, potentially transforming healthcare diagnostics.

C. Medicine Reminder

Irawan et al. [4] addressed the difficulties posed by leprosy, also known as Hansen's disease (HD) or Morbius Hansen's disease (MHD), a condition caused by the bacterium Mycobacterium leprae. Recognizing symptoms of leprosy can be challenging, sometimes taking 20-30 years to identify. Although multidrug therapy (MDT) and anti-inflammatory treatments have greatly improved health outcomes for leprosy patients, the lengthy medication process requires disciplined treatment management to prevent permanent disability or deformity. To streamline this process, the researchers propose the use of smartphone technology, particularly Android-based applications, to assist patients with medication management, supervision, and reminders. Furthermore, cloud-based technology is envisioned as a communication and information medium for all stakeholders involved in leprosy patient care. The application's functionality and design were evaluated using the User Experience Questionnaire (UEQ) and black-box testing, which found no issues with the app's functionality. The UEQ results demonstrated a positive reception, with the application scoring highly on novelty, stimulation, efficiency, reliability, perspicuity, and beauty. With the highest rating received for efficiency, the application is deemed as highly efficient and user-friendly according to UEQ results.

D. Location Recommendation with Google Maps API

A model for travel recommendations is proposed [5] that is implemented in an Android mobile application. The model uses three criteria to recommend tourist attractions: ticket prices, weather, and rating. The authors use a simple additive weighting method to calculate the weights for each criterion. The authors evaluated their model on a dataset of tourist attractions in Indonesia using the Google Places API. The results show that the model can recommend tourist attractions that meet the criteria desired by tourists.

E. Existing Applications

WebMD [6] stands as a leading source of reliable health information, having registered around 127 million unique visitors in Q1 2020. It provides a health dictionary, healthy lifestyle advice, latest news on health, drug and supplement information, among other features. Its unique rule-based symptom checker engages users by accepting symptom inputs and checking these against a built-in decision tree. It poses follow-up questions based on initial symptoms, generating a list of potential diseases and associated treatment advice.

Symptoma [7] a digital health assistant, leverages machine learning capabilities to provide symptom checks and disease diagnoses. It is a product of 16 years of research by medical doctors and data scientists and boasts a remarkable accuracy rate exceeding 95% in predicting 20,000 diseases. Users input symptoms, answer follow-up questions, and receive a list of matching diseases sorted by probability. This interactive process, which adjusts the confidence level with each new symptom, facilitates more accurate diagnoses.

OpenEMR [8] is a comprehensive, free-to-use electronic health records web application. It is ONC certified and provides full integration of health records, practice management, scheduling, electronic billing, and more. Users can record demographics, medical problems, allergies, and medication details, supporting its use on a variety of platforms.

Medisafe [9] is a free mobile application that primarily serves as a medication reminder. The app enables users to create personalized reminders for each medication, sends refill reminders, and offers vital drug interaction warnings. Users can also track their disease progression, allowing for better monitoring and management of their health conditions.

Getdoc [10] is a mobile application designed to connect users with healthcare providers such as specialists, doctors, pharmacies, etc. It covers Thailand, Malaysia, and Singapore, with an extensive database of healthcare providers. Users can search and filter by name, location, specialty, and country, make appointments online if the provider is registered, or get contact details for unregistered providers. This makes healthcare access and navigation more efficient for its users.

F. Comparison with Existing Applications

The proposed platform distinctively amalgamates features from multiple established health applications, while adding its unique enhancements. Similar to WebMD, the proposed system offers a symptom checker, employing a Random Forest model instead of a rule-based system, which may enhance the accuracy of diagnoses. Following Symptoma's approach, the proposed system utilizes machine learning and enhances user engagement by integrating an OpenAI API-powered chatbot. While OpenEMR's feature of recording medical data is mirrored in the proposed system, a more interactive Medication Reminder feature, like Medisafe, has been offered in our proposed system to ensure medication adherence. It also integrates Google Maps API, akin to Getdoc's location-based services, but it recommends nearby hospitals and clinics. Essentially, the proposed Flask-based platform combines various robust features from existing applications and elevates them, resulting in an efficient and comprehensive healthcare management tool. A comparison of existing applications is provided in Table 1.

Table 1. Comparison of features between existing applications

Application/ Feature	Symptom Checker	Personal Health Record	Medicine Reminder	Appointment Scheduling	Artificial Intelligence	Doctor Recommendation
WebMD	√	√		√		
Symptoma	√				√	
OneTouch EMR		√		√		
MediSafe			√	√		
Getdoc				√		√
HealthyMe (proposed system)	√	√	√		√	

III. RESEARCH METHODOLOGY

A. System Development Life Cycle

Waterfall Model of the System Development Life Cycle (SDLC) [11] has been employed to guide the software development process, proceeding in a linear sequence through requirement analysis, system design, implementation, testing, deployment, and maintenance. Initially, extensive research was conducted on similar healthcare applications to ascertain necessary requirements, including programming language selection, target audience, and system functionalities. Following this, the system design phase encompassed hardware and software requirements, as well as the design of prototypes, databases, and interfaces using various diagrams. The implementation phase witnessed the construction of the application using Python and Flask, HTML, CSS, JavaScript for the user interface, and SQLite for data management, with features like a machine learning model for disease diagnosis, personal health records, and medicine reminders. The system testing phase ensured all functional and non-functional requirements were met, with each test case scrutinized and passed only if working as intended.

B. OpenAI API

The OpenAI API was employed to enable the integration of powerful AI functionalities into the application. The OpenAI GPT-3.5 model was utilized to enhance the medical portal by offering drug interaction checks, providing an AI-driven, interactive, and informative experience for the users. OpenAI was chosen because of its advanced language processing capabilities, proven to be capable of generating human-like text based on provided input. The OpenAI API-powered chatbot acts as a virtual healthcare assistant, engaging in natural language conversations with users. Users can provide inputs like, symptoms to predict disease which system will retrieve medical history from database to help act as a diagnosing factor , find nearby hospital or clinics, medicine name, side effects, and treatments , and available in multiple languages although only Malay and English is only shown in this paper. The ConversationHistory which saved user conversation with chatbot will be retrieved to help the chatbot stay in context.

C. Data Preprocessing

Data preprocessing is an integral part of machine learning pipeline, ensuring that our data is appropriately cleaned and transformed for use in our models [12]. In this project, we employed several preprocessing techniques designed to convert categorical data to numerical format and handle missing data. Initially, we checked the dataset for null values and removed any detected. Next, categorical encoding was carried out on the target variable 'prognosis' using Label Encoding, transforming it from categorical data into a numerical format suitable for machine learning algorithms. This stage of preprocessing allowed us to ensure data consistency and optimize the structure of the data for our machine learning models, thereby enhancing their overall effectiveness and improving the accuracy of the models' subsequent predictions.

D. Random Forest

The Random Forest algorithm was used as the primary predictive model in our project. This decision tree-based ensemble learning method provides high accuracy, robustness, and ease of use. It operates by constructing multiple decision trees and outputting the mode of the classes for classification or the mean prediction for regression [13].The algorithm was chosen for its ability to handle a large number of inputs, its robustness to outliers, and its capacity for feature importance estimation of disease. It effectively managed the various data features of our project, providing excellent predictive power and aiding in feature selection. The Random Forest model-based symptom checker in our proposed system analyzes user-provided symptoms to suggest potential diseases, providing users with a preliminary diagnosis. The underlying principles of the Random Forest model shows its advantages in accurately predicting and classifying symptoms.

E. Google Maps API

The Google Maps API was incorporated into our project as a tool to provide geolocation services and to enhance the functionality of our OpenAI chatbot. The Google Maps API enables our system to recommend proximal hospitals based on the user's location. By leveraging geolocation services, users can easily access relevant healthcare facilities in their vicinity, ensuring prompt medical attention and reducing the time taken to reach appropriate healthcare providers. Google Maps JavaScript API and Places API were used to fetch information about nearby hospitals and clinics. The chatbot takes user input in the format: "Find hospitals near {location} within {radius} km". If no radius is specified, it defaults to 5000 meters. The user input is parsed to extract the location and optional radius, which are used to make a request to Google's Places API. The chatbot responds by listing the nearby hospitals and clinics along with their vicinity details. The sendMessage() function is executed whenever a message is sent. If the user's message starts with "find hospitals near", it treats the rest of the message as a location name and attempts to find hospitals near that location using Google's Geocoding and Places APIs. If the geocoding operation fails, it notifies the user that the location could not be found. If the user's message does not request nearby hospitals, it is sent to a backend endpoint (/api) using the Fetch API, and the response from the endpoint is added to the chat box as a bot message.

F. Dataset

The dataset was retrieved from Kaggle [14]. It consists of a Training.csv for training the machine learning model and Testing.csv to test the performance of the machine learning model after the completion of training process. The training dataset consist has 4920 rows of data and 133 columns, which 132 is symptoms and last column is prognosis, which has 42 unique disease.

G. Support Vector Machine

A Support Vector Machine (SVM) was used as one of the primary machine learning models in this study. SVM is a supervised learning model that uses classification algorithms for two-group classification problems. Its benefits extend to diverse applications, including the prediction of travel insurance purchases [15] and disease prognosis. After providing an SVM with a set of training examples, each marked as belonging to one or the other of two categories, the SVM model constructs a model that assigns new examples into one category or the other. In this study, the SVM model was used to predict diseases based on the symptoms inputted. A multi-class classification approach was adopted as the prognosis contained multiple categories. The SVM model was trained on the training data and tested on the test data. The model achieved a perfect accuracy score of 100%, indicating that it was able to correctly classify every instance in the testing dataset.

H. Convolutional Neural Network (CNN)

A CNN is a type of deep learning model that is typically used for image processing, but in this case, it was employed for sequential data. CNNs are proficient at identifying patterns in the input data, which makes them suitable for tasks like disease prediction. For the implementation of the CNN model, the input data was first reshaped to make it compatible with the CNN. The network architecture was defined, and then the model was trained using the training data for up to 300 epochs. Early stopping was also implemented to prevent overfitting of the model on the training data. Like the SVM model, the CNN also achieved an accuracy score of 100% on both the training and testing datasets.

I. Fernet

Our application employs Fernet, a symmetric encryption scheme, to ensure secure handling of user conversation histories at backend. Using Fernet's AES-based cryptography and HMAC for verification, the user's conversation history is decrypted upon receipt of a POST request. Every subsequent user input and AI-generated response is encrypted using Fernet before being stored in the database. It fetches the encrypted conversation history from the received JSON data and decrypts it is using the Fernet cipher suite, which was instantiated with a previously generated secret key. Any incoming message that is prefixed with "enc_" is considered encrypted and is decrypted appropriately before passed to OpenAI Server when user send a message. The encrypted information can only be accessed with the secret key.

IV. RESULTS AND DISCUSSIONS

This section provides a visual representation of the project's architecture, structure, and components. Diagrams help in comprehending the system's various elements, their interactions, and their relationships. By visualizing the design, project teams can proactively address these risks, plan mitigation strategies, and make informed decisions to minimize the impact on the project's success.

Figure 1 illustrates the movement of data within the system. It begins from user registration, navigating through login, profile navigation, and ends with logout. In between, it shows the management of personal data, such as additions, edits, and deletions in the medical history and allergy records. It also indicates how users can set up and modify medication reminders. Symptom checker are using Random Forest Model and Chatbot will be connected to OpenAI Server with OpenAI API and Google Server with Google Maps API which will also retrieve medical details from Medical History to help diagnose disease and content from Conversation History to stay in context. Content of conversation is also encrypted before sending to database and decrypted when retrieving from database.

Figure 2 presents the sequence of interactions for various tasks like registration, login, and navigation on the main page. It also shows how users can add, edit, and delete entries in the medical history, allergy, general information, and medicine reminders. Additionally, the diagram explains how the symptom checker are used for symptom analysis and chatbot for interaction. Activity 4.0 is chatbot follow by inquire Medical Information 4.1, and view responses and follow up questions for 4.4.1.

Figure 3 provides a comprehensive graphical representation of the entire system, effectively illustrating the interplay between various elements that make up the medical diagnosis platform. At the forefront is the User Interface, which is designed to be interactive, intuitive, and user-friendly. This is the primary point of contact between the user and the system, where symptom input is received, and diagnosis results are displayed. Behind the scenes, the user data is securely stored and managed in the Database. This information includes the user's medical history, general information, allergy, and the conversation history for chatbot. This system is designed to be robust, ensuring data integrity and security while allowing for efficient data retrieval and updates as the conversation history is also encrypted. The core component of the system is the Machine Learning Model that performs the vital task of symptom analysis and disease prediction. It integrates Random Forest Model and Predict top 5 disease algorithms to provide accurate and timely disease predictions based on the symptoms input by the user. Supplementing the User Interface is the Chatbot, which provides an alternative interaction mode for users. It allows user queries and provides intelligent responses, thereby enhancing user engagement and can also retrieve patient medical history when asked to diagnosis disease. Lastly, the system architecture highlights the integration of External APIs such as the OpenAI API and Google Maps API. The OpenAI API powers the Chatbot, enabling advanced language understanding and generation, while the Google Maps API is utilized for geographical data handling, for instance, to list several nearest hospitals given an address.

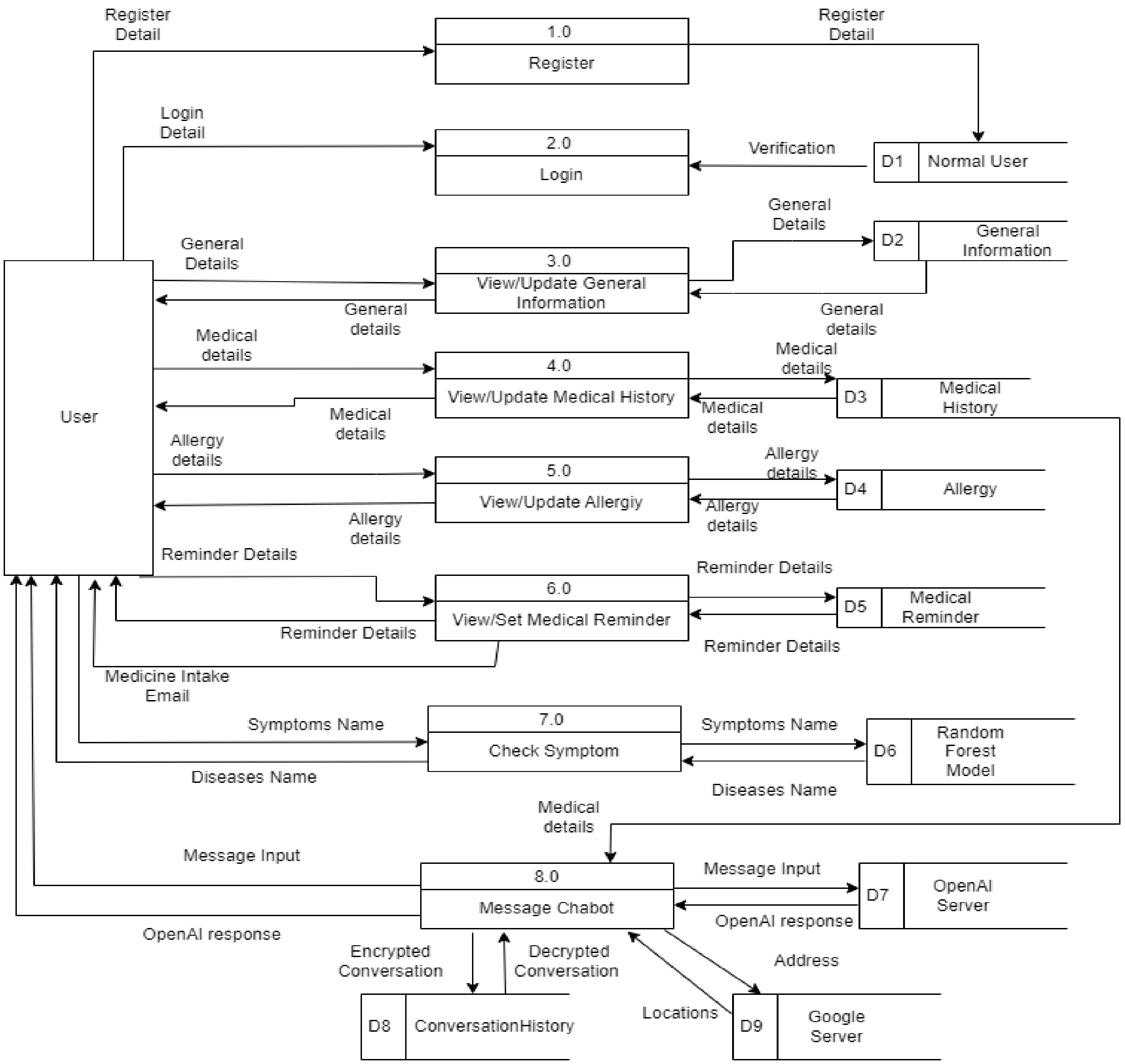

Figure 1. Data flow diagram

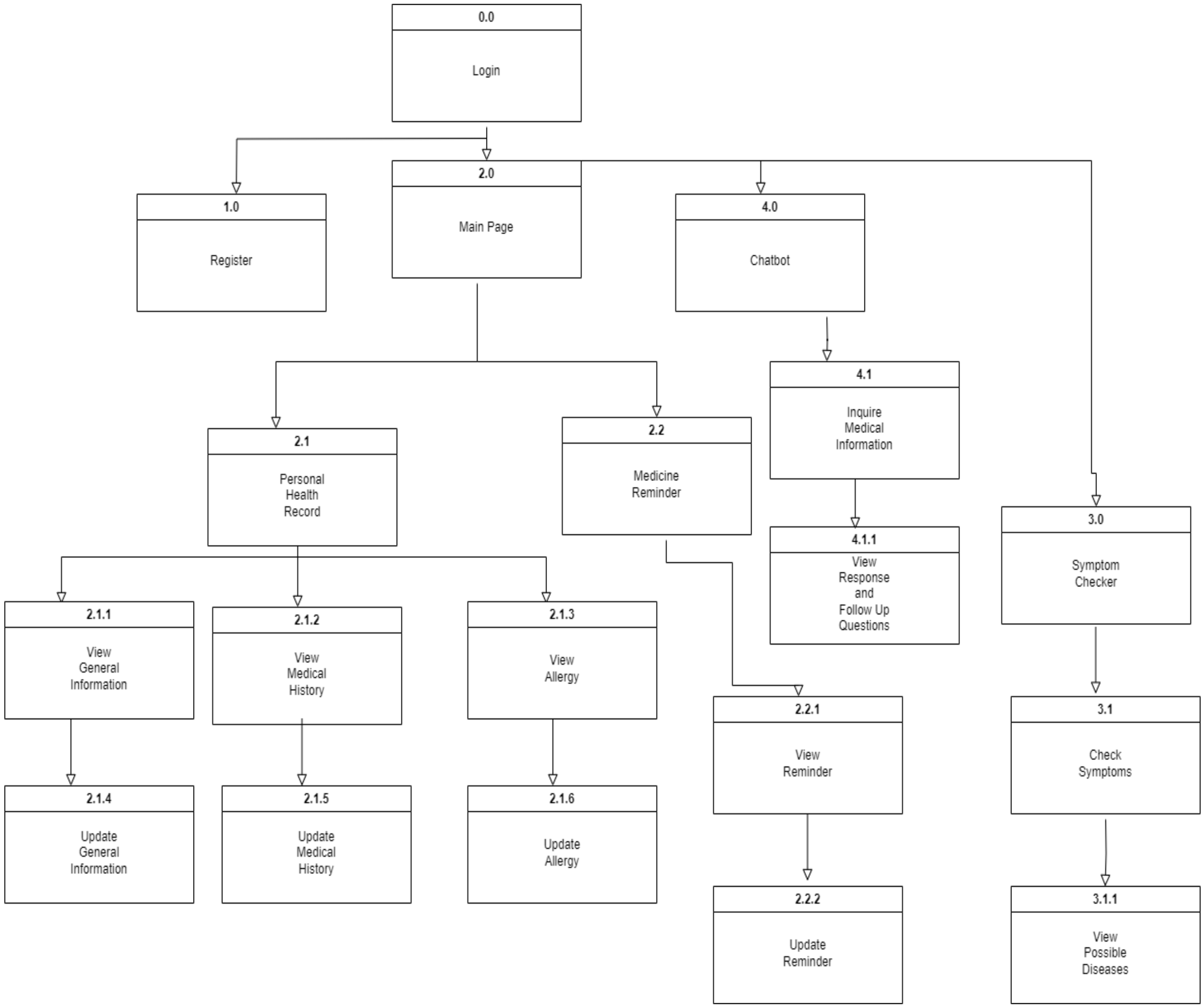

Figure 2. Dialogue diagram

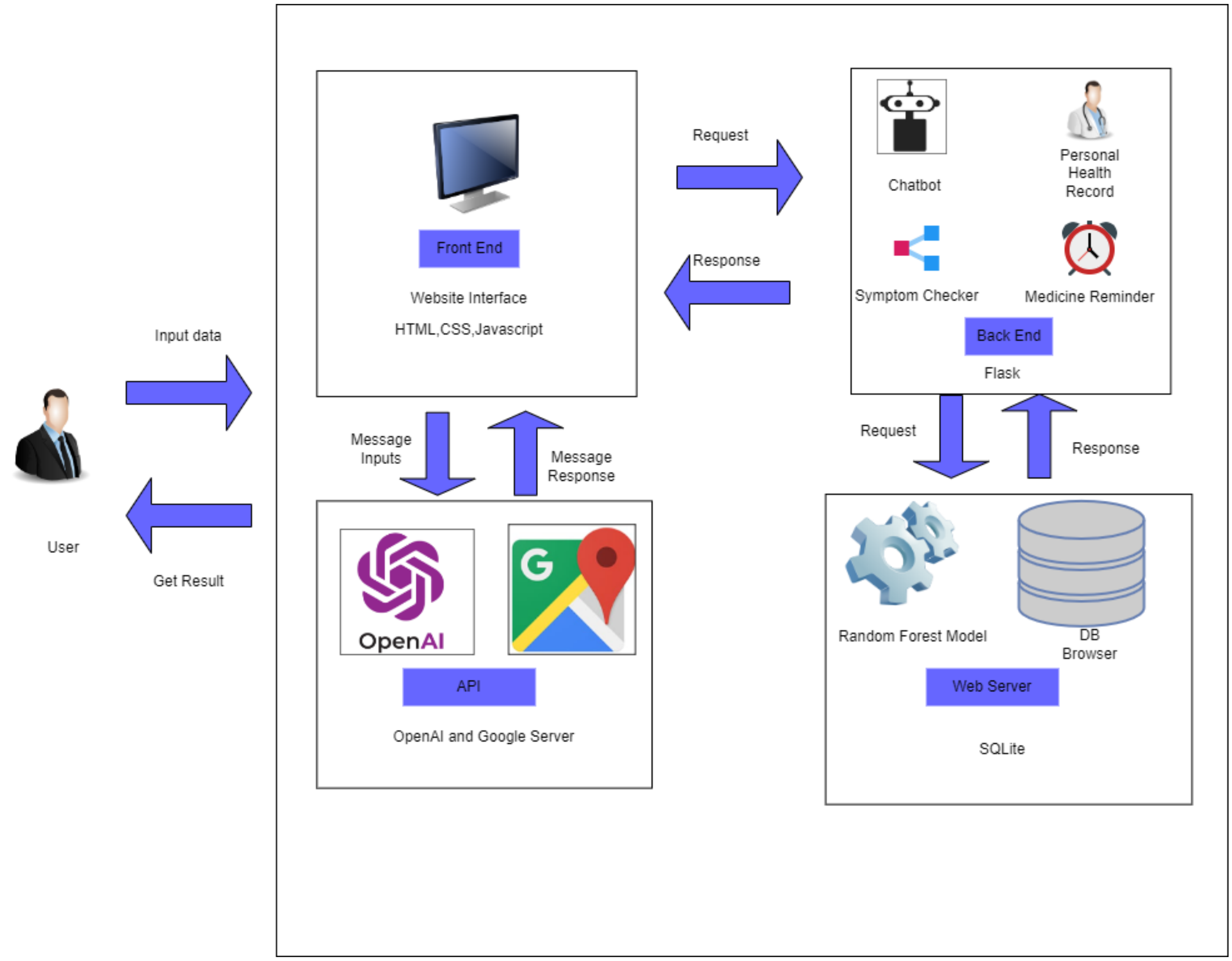

Figure 3. System architecture of the proposed system

V. SYSTEM PREVIEW

The system preview section provides an overview and glimpse into the key components, functionalities, and architecture of the proposed system. With the rapid advancements in technology, there is an increasing need for innovative solutions in the healthcare sector. The proposed system addresses this need by harnessing cutting-edge technologies like machine learning, natural language processing, and geolocation services, crafting a personalized healthcare experience. This system preview aims to demonstrate the innovation within the proposed solution and emphasize its potential to transform how individuals access healthcare services. By combining advanced algorithms, AI-powered chatbots, and geolocation services, our system offers a comprehensive and personalized approach to preliminary disease diagnosis and hospital recommendations.

A. Main Page

Post-login, users land on the main page, as displayed in Figure 4, hosts navigation to various other pages such as Symptom Checker ML, Symptom Checker API, HealthyMe Chatbot, General Information, Medical History, Allergy, Medicine Reminder, and Log Out, and have a disclaimer that the website does to substitute professional medical advice and only for informational purposes.

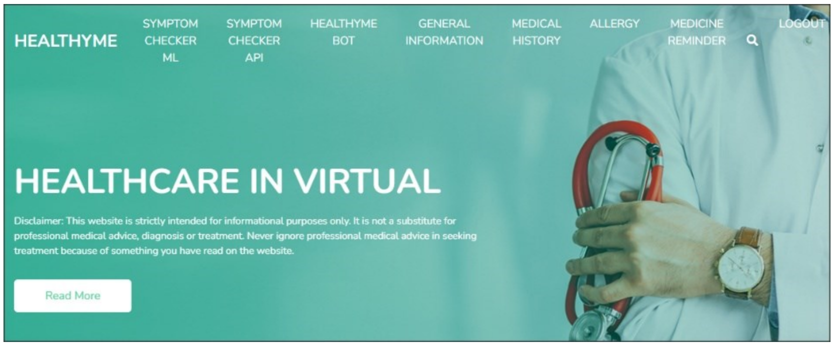

Figure 4. Main page

B. Medical History & General Information

This feature allows users to record, view, and manage their personal medical history. They can add a new record, view saved records, edit existing entries, or delete records as needed, providing comprehensive control over their medical history data. Figure 5 illustrates the Medical History Page. Users can enter and update personal details on the 'General Information' page. All entries that are validated, are stored in the database and made available for viewing.

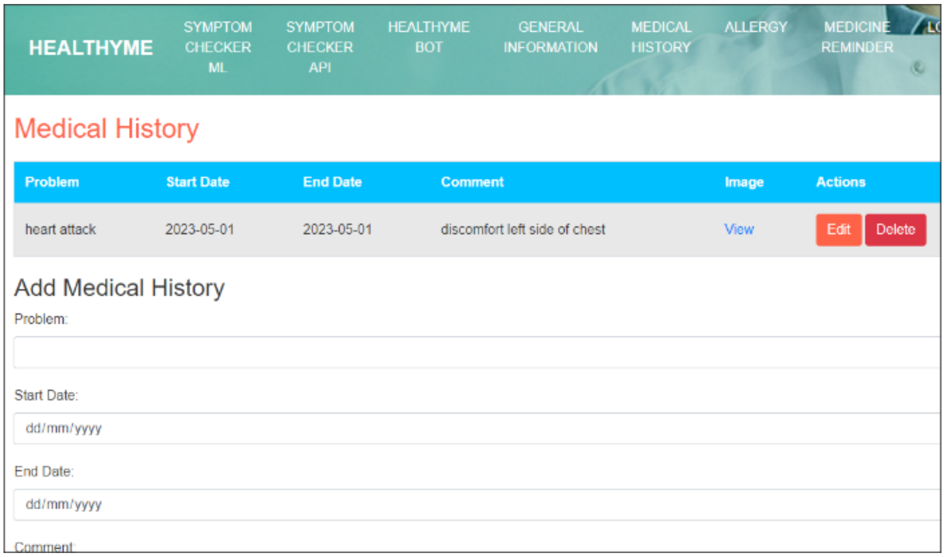

Figure 5. Medical history page

C. Allergy

Figure 6 presents the Allergy Information page, that enables users to maintain a record of their allergies, related reactions, and their severity. Users can add, view, edit, and delete allergy entries, providing a comprehensive repository for allergy management.

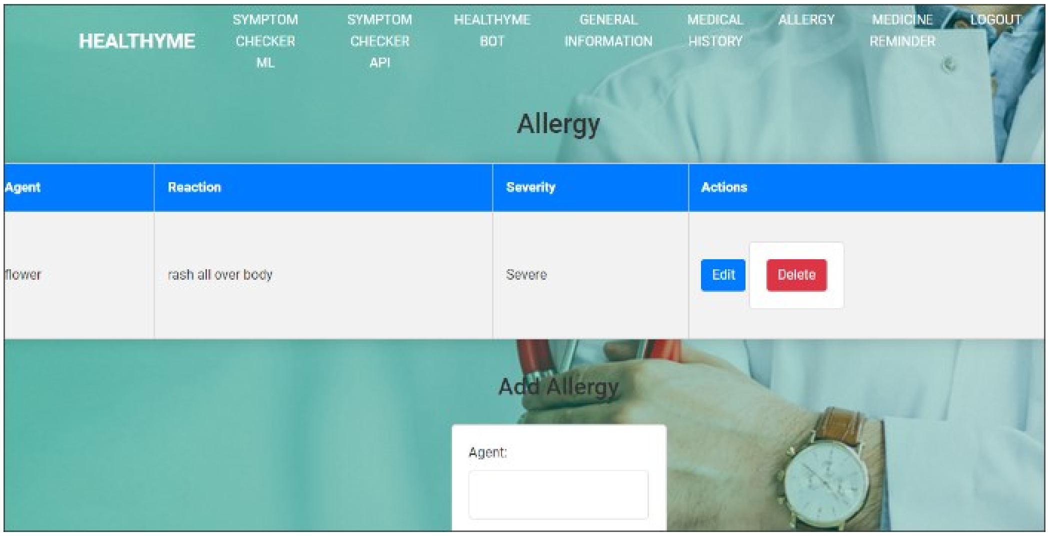

Figure 6. Allergy information page

D. Medical Reminder

This Medicine Reminder as illustrated in Figure 7, assists users in managing their medication regimen. Users can add, edit, or delete medicine reminders which are displayed in a table format. If no reminders are set, a message stating "No medicine reminders found" is displayed. User will receive email notification based on the information added according to time, and repeat days.

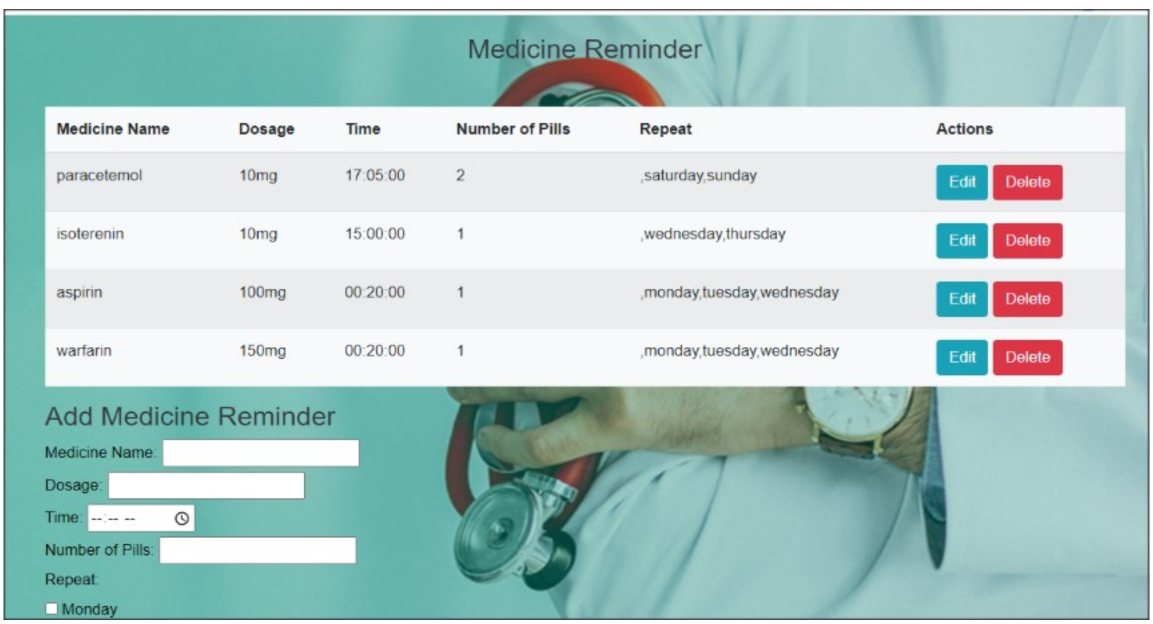

Figure 7. Medicine reminder

E. Symptom Checker with Machine Learning

System Checker utilizes Random Forest Model to predict potential illnesses based on the symptoms selected by the user. After symptom selection and prediction initiation, the system presents the top five potential diseases associated with the provided symptoms. Figure 8 shows the page view of the system checker.

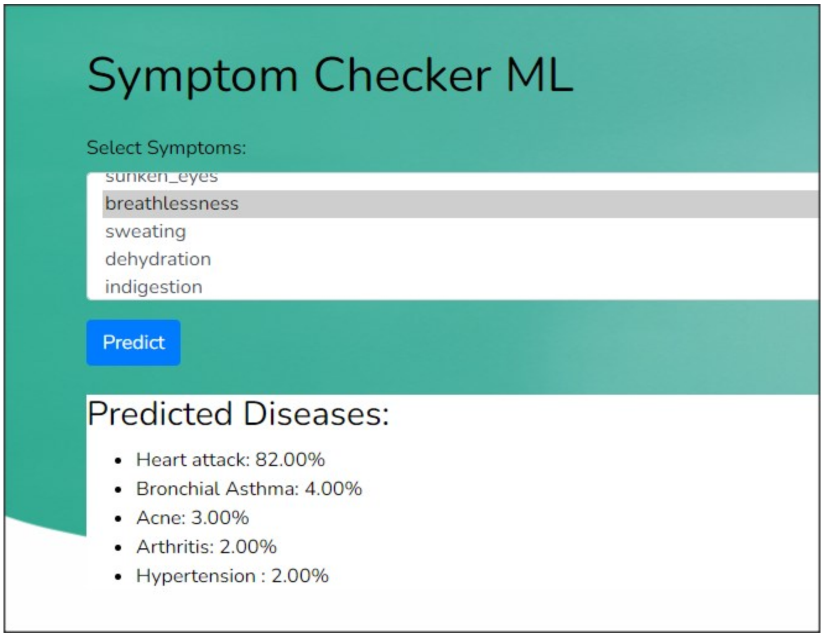

Figure 8. System checker

F. OpenAI API Chatbot

The OpenAI API Chatbot, named as HealthyMe bot, leverages the OpenAI API to understand and respond to health-related queries. Features such as disease diagnosing via symptoms, find nearby hospitals, provide information about medicines, and more. Its capabilities are enhanced through Javascript customization such as Google Maps API, allowing for a robust and interactive user experience. All of the user and bot messages are stored in a "conversationHistory" array, which is used to maintain the context of the conversation. When the user sends a message, it's added to the chat box and "conversationHistory". Figure 9 presents the OpenAI API Chatbot page.

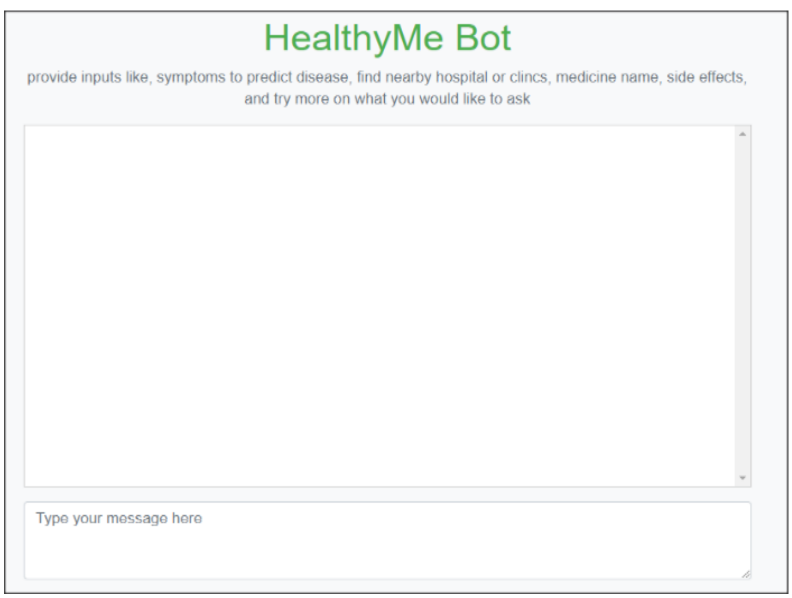

Figure 9. OpenAI API chatbot

The system can retrieve medical history of user from database to act as a consideration factors if it exists and to be appended in the prompt to OpenAI server when predicting diseases based on symptoms given. Figure 10 shows the predicting result of the symptom integrated with the medical history stored in the database.

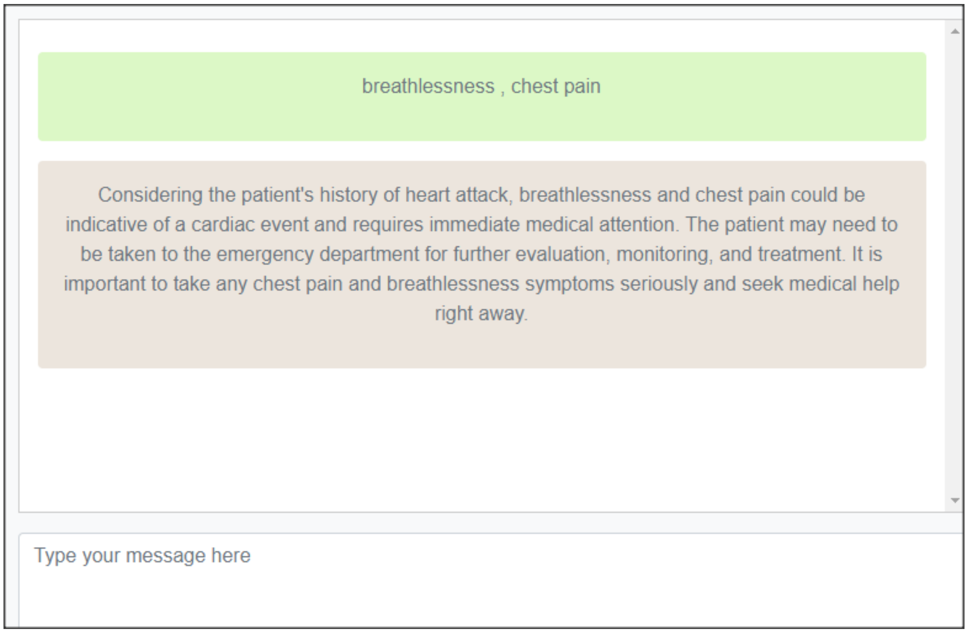

Figure 10. Symptom diagnosing integrated with medical history

The system incorporates Google Maps' Geocoder and Places APIs to efficiently locate nearby hospitals or clinics based on user input. By geocoding the user's provided location, the system then performs a targeted search within a 5000-meter radius to identify and recommend relevant healthcare facilities. Figure 11 displays the result of finding nearby hospitals.

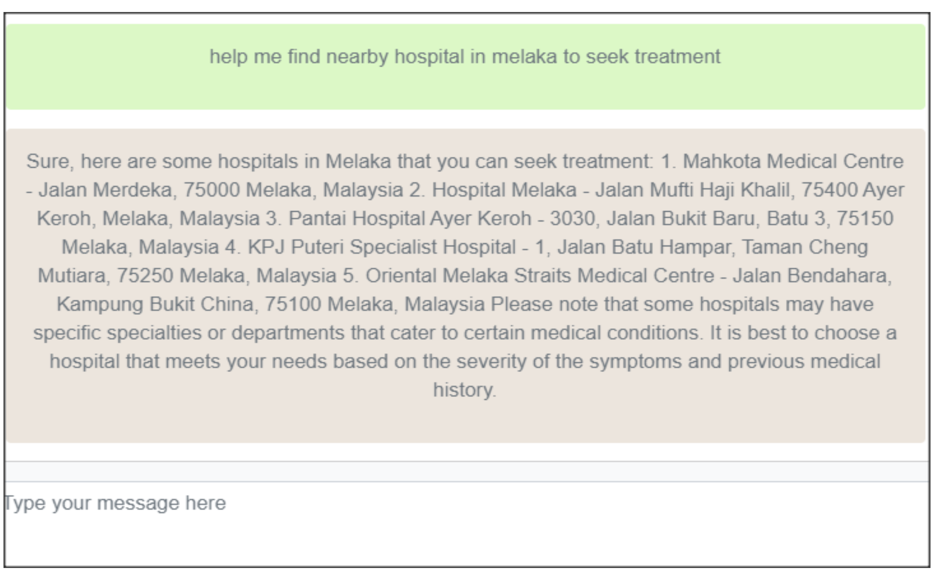

Figure 11. Find nearby hospitals

Figure 12 shows the Medicine Information Inquiry page. Users can inquire about medicine information, including details about specific medicine names, their associated details, and potential side effects.

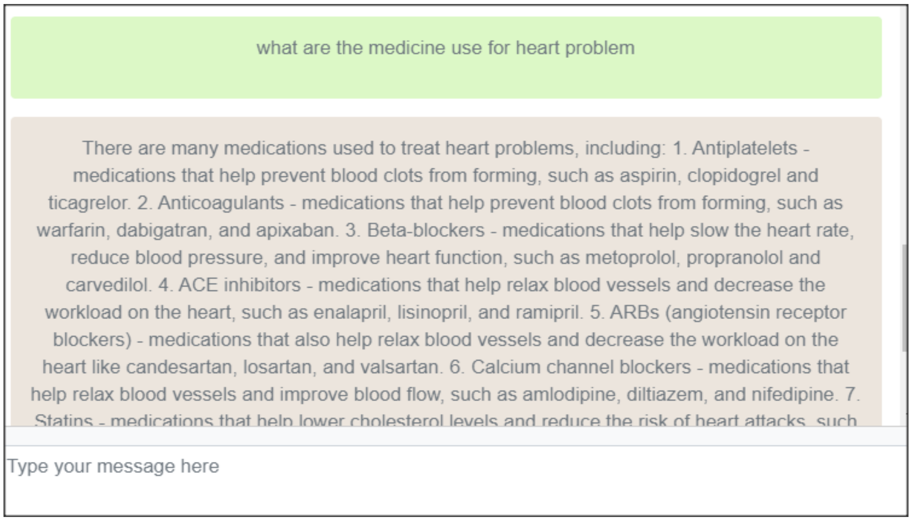

Figure 12. Medicine information inquiry

G. Quick Symptom and Disease Checker

Quick Symptom Checker offers a more direct, form-based approach to gathering information about diseases based on symptoms. The user enters their symptoms and receives a list of possible conditions as shown in Figure 13.

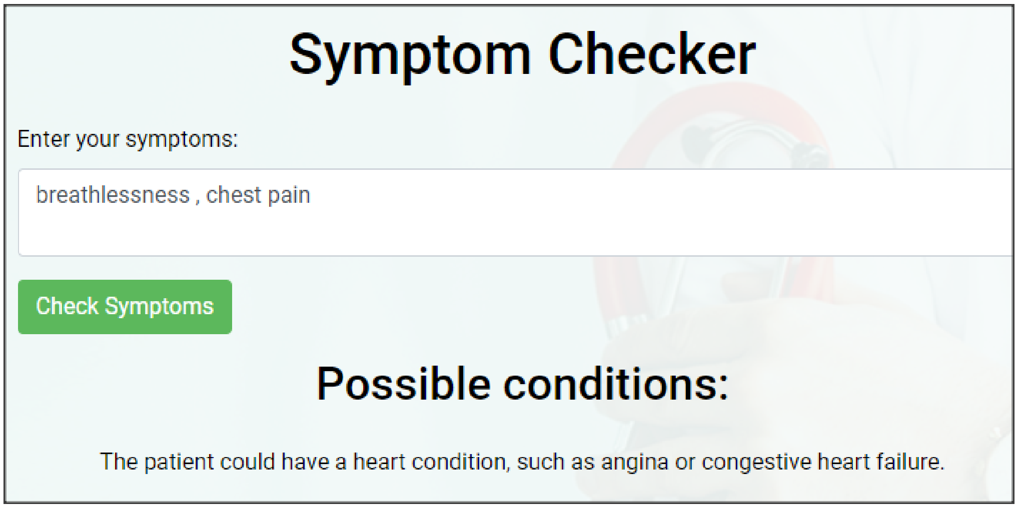

Figure 13. Quick symptom checker

Upon entering a disease name, users can access comprehensive and detailed information regarding the specific disease, enabling them to gain a deeper understanding of its characteristics and implications, as presented in Figure 14.

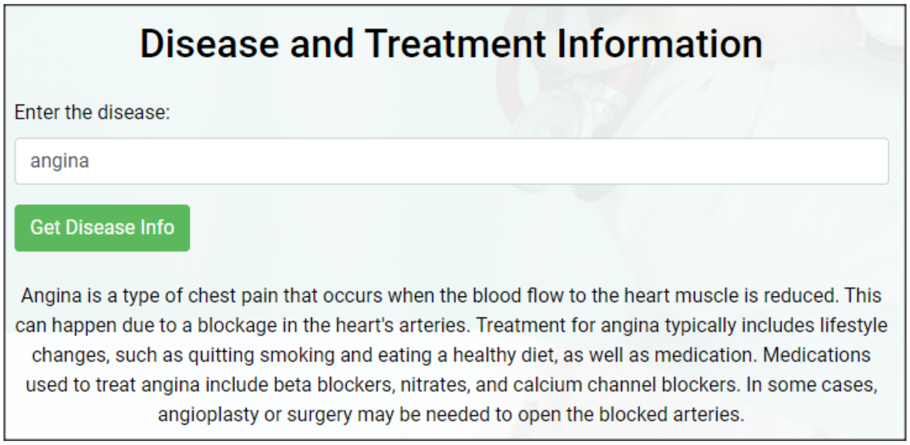

Figure 14. Quick disease inquiry

VI. EXPERIMENTAL RESULT

In this study, the focus is on the application of three different machine learning models to predict disease based on symptom data: Random Forest, Support Vector Machines (SVM), and Convolutional Neural Network (CNN).A dataset composed of 4920 instances, each characterized by 132 different symptoms and the corresponding diagnosis, was utilized. The dataset was initially processed to ensure it did not contain any missing values. All data was successfully loaded without any null values identified across the various symptoms or in the prognosis column. The prognosis column was then label-encoded to convert the categorical data into numerical format, facilitating the application of machine learning algorithms. For experimental validation, the dataset was split into training and testing sets, with a ratio of 80:20. The training dataset consisted of 3936 samples, while the test dataset had 984 samples. Each of the three machine learning models was trained on the training set and subsequently evaluated on the test set.

The Random Forest Classifier was set with 50 estimators and a maximum depth of 5. The model was trained and tested, yielding an accuracy of approximately 98.65% on the training data and 97.62% on the test data. An SVM model was next employed, which achieved perfect accuracy both on training and testing sets, with scores reaching 100%. Finally, a 1D Convolutional Neural Network model was constructed and trained. After reshaping the input data for compatibility with the CNN model and setting up the model architecture, it was trained for up to 300 epochs with early stopping implemented. Similar to the SVM model, the CNN model achieved an accuracy of 100% on both the training and testing datasets. Figure 15 shows the comparison of accuracy between models.

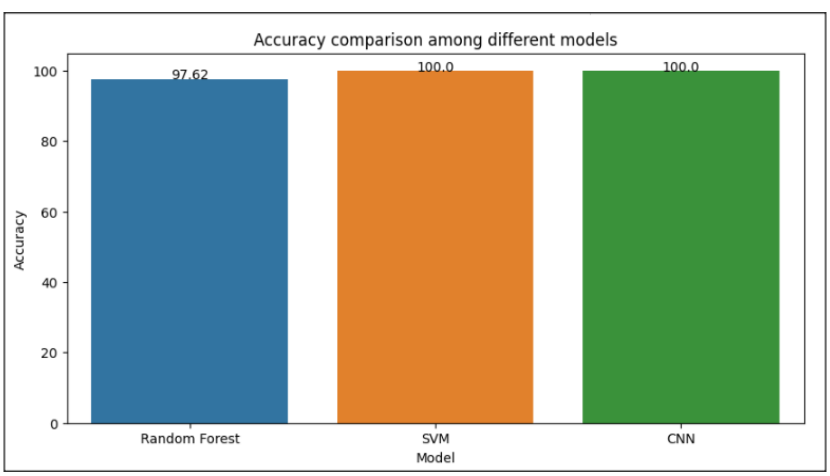

Figure 15. Accuracy comparison among different models

Upon analysis of precision, recall, and F1-score as shown in Figure 16, the Random Forest model demonstrated scores of approximately 0.99 for precision and recall, and 0.98 for the F1-score. Conversely, both SVM and CNN models scored perfectly that is 1.0 across all these metrics.

The model's ultimate objective was to predict the top 5 diseases based on provided symptoms. A dictionary was created to map symptoms to their corresponding index, which allowed for encoding of input symptoms into a numerical form that can be used by the model. The function 'predictTop5Diseases' takes a string of symptoms separated by commas, encodes them, and uses the model to predict the top 5 probable diseases. Finally, the mapping of diseases from numerical back to their original form was saved for future reference in the system, allowing for the interpretation of the model's predictions.

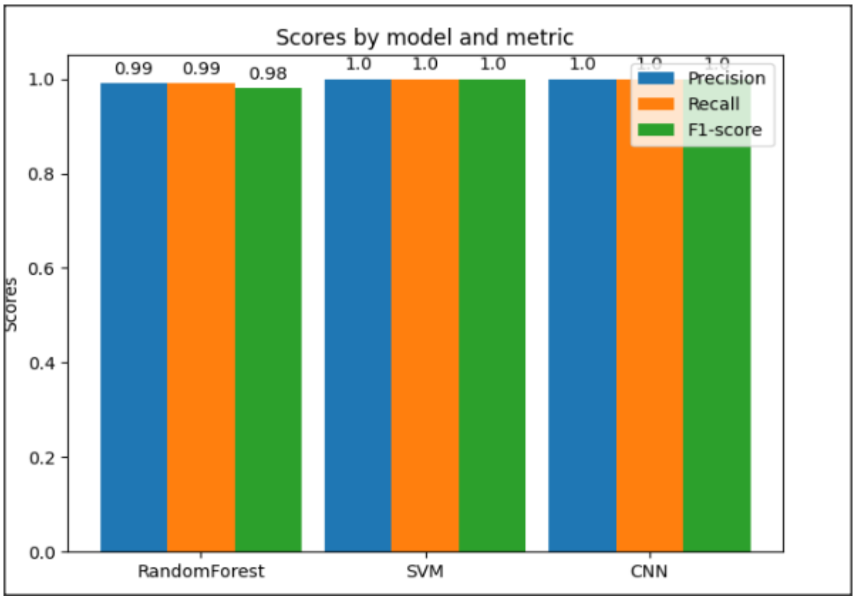

Figure 16. Scores by model and metric

In terms of interpretability, the RF model stood out. It provides a measure of feature importance, which indicates the relative contribution of each symptom to the prediction of a disease. This transparency can be particularly useful in a healthcare setting, where understanding the basis of a model's prediction can be as important as the accuracy of the prediction itself. In contrast, both the SVM and CNN models are typically seen as 'black box' models, making predictions based on complex transformations of the data that can be difficult to interpret.

To showcase the scenario, the models were given two symptoms, breathlessness and chest pain, and asked to predict the top five diseases. The RF model predicted a 82% chance of a heart attack, while the SVM model predicted a 69% chance. Although the SVM model had overall higher performance metrics, the RF model provided a more accurate prediction in this specific scenario, further illustrating the balance between performance and interpretability as shown in Figure 17 and 18.

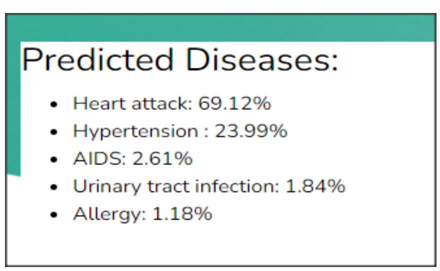

Figure 17. SVM Model top 5 predictions given symptoms

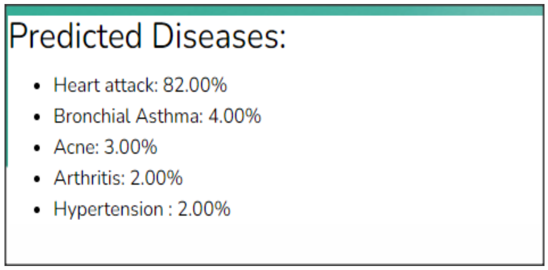

Figure 18. RF Model top 5 predictions given symptoms

In terms of interpretability, the RF model stood out. It provides a measure of feature importance, which indicates the relative contribution of each symptom to the prediction of a disease. This transparency can be particularly useful in a healthcare setting, where understanding the basis of a model's prediction can be as important as the accuracy of the prediction itself. In contrast, both the SVM and CNN models are typically seen as 'black box' models, making predictions based on complex transformations of the data that can be difficult to interpret. To showcase the scenario, the models were given two symptoms, breathlessness and chest pain, and asked to predict the top five diseases. The RF model predicted an 82% chance of a heart attack, while the SVM model predicted a 69% chance. Although the SVM model had overall higher performance metrics, the RF model provided a more accurate prediction in this specific scenario, further illustrating the balance between performance and interpretability.

Upon testing, the chatbot was able to successfully recognize the user's request for nearby hospitals and clinics which this feature is unable to work in original ChatGPT website. When provided with a location and an optional radius, the chatbot returned a list of nearby hospitals and clinics and it can also stay in context of the conversation, for example in Figure 19, mentioned by disease instead of directly saying heart attack which is retrieved from the previous user conversation history and it also allows user to follow up with a question about the hospital suggested by the chatbot. If no radius was provided, the chatbot defaulted to a 5 km radius around the specified location. Since it is integrated with OpenAI API, it has Natural Language ability and can understand and respond with other languages as shown in Figure 20. Additionally, Fernet was implemented to encrypt and decrypt the content in Conversation History for both user and chatbot as shown in Figure 21 for privacy purpose. When a user input does not match the expected pattern to find hospitals and clinics, the chatbot processes the input as a normal chat message, showing flexibility in managing different user requests. The function getNearbyHospitals is flexible enough to search both for hospitals and clinics. This could be extended to other types of places in the future.

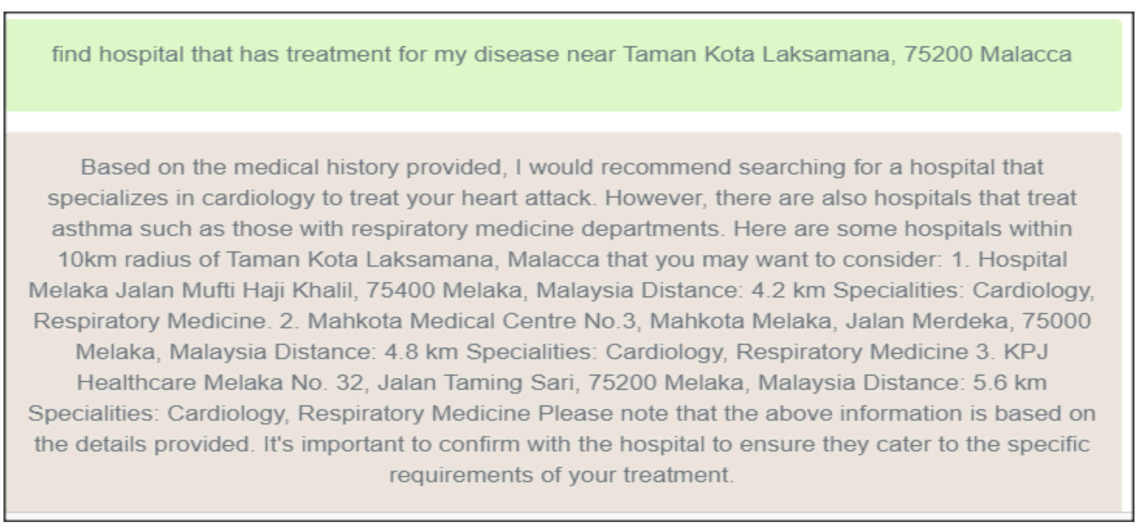

Figure 19. Example of query of find hospital nearby a given address

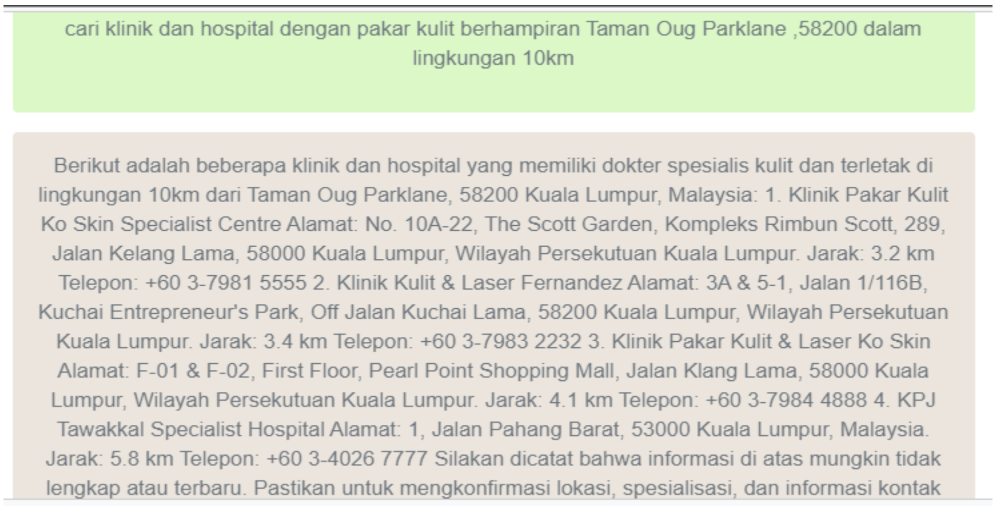

Figure 20. Example of query of find hospital and clinics in Malay language

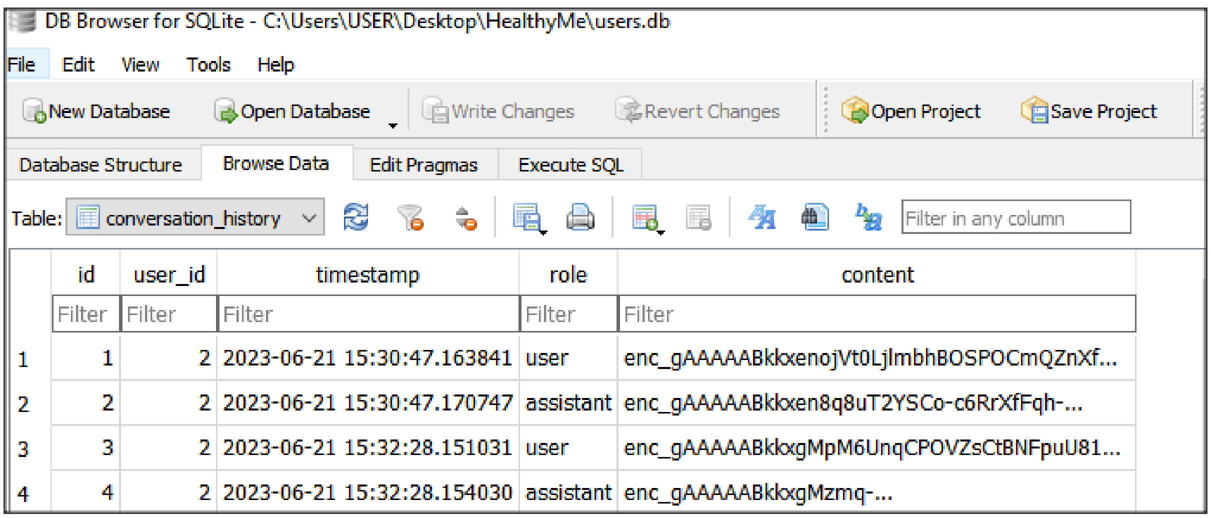

Figure 21. Encrypted user conversation content

VII. CONCLUSION

This paper introduces a comprehensive approach for symptom diagnosis and hospital recommendations using AI and Location Services to enhance the personalized healthcare. The proposed system, named as HealthyMe, is a personal healthcare system which epitomizes the intersection of healthcare and technology, empowering users to take charge of their health data. It offers a comprehensive health record system that improves the quality of healthcare and encourages proactive health practices. The system enables users to effectively manage personal health information, track health patterns over time, and make informed decisions. Advanced features, such as the AI-driven Symptom Checker and Medicine Reminder, elevate this system from a simple record tool to a personal health assistant. The accuracy of the proposed system on the test dataset using the training model is 97.62%. The system's potential for enhancements includes integration with telemedicine platforms, expanding the symptom checker dataset with contemporary dataset if available, and refining the AI for more natural interactions. In essence, the proposed system uses artificial intelligence to foster efficient, user-centric health management, promoting informed decision-making and contributing to a healthier society.

ACKNOWLEDGEMENT

A version of this paper was presented at the third International Conference on Computer, Information Technology and Intelligent Computing, CITIC 2023, held in Malaysia on 26th-28th July 2023.

REFERENCES

[1] R. Kannan, I. S. Rosdi, K. Ramakrishnan, H. R. Abdul Rasid, M. H. L. Mohamed Rafy, S. Yusuf & S. N. A. Mohd Salamun, "Leveraging Business Data Analytics and Machine Learning Techniques for Competitive Advantage: Case Study Evidence from Small Businesses", International Journal of Management, Finance and Accounting, vol. 2, no. 1, pp. 73–87, 2021.

[2] L. Kaboutari-Zadeh, A. Azizi, A. Ghorbani & A. Azizi, "Designing and evaluating a mobile personal health record application for kidney transplant patients", Informatics in Medicine Unlocked, vol. 30, 2022.

[3] C. N. Villavicencio, J. J. Macrohon, X. A. Inbaraj, J. H. Jeng & J. G. Hsieh, "Development of a Machine Learning Based Web Application for Early Diagnosis of COVID-19 Based on Symptoms", Diagnostics, vol. 12, no. 4, pp. 1 – 30, 2022.

[4] P. L. T. Irawan, C. B. S. Hartanto & O. H. Kelana, "Medicine Consumption Reminder and Monitoring Application for Patients with Leprosy Disease", Journal of Community Practice and Social Welfare, vol. 2, no. 1, 2022.

[5] R. Wahdiniwaty, E. B. Setiawan, F. Auliardi & D. A. Wahab, "Application Model for Travel Recommendations Based on Android", IJNMT (International Journal of New Media Technology), vol. 6, no. 1, 2019.

[6] WebMD, WebMD Symptom Checker, WebMD LLC, https://symptoms.webmd.com/ (accessed April 4, 2023)

[7] Symptoma, Welcome to Symptoma. Symptoma, GmbH, https://www.symptoma.com/ (accessed January 11, 2023)

[8] OpenEMR. Fully Working OpenEMR 7.0.1 Demo, OpenEMR Foundation, Inc., https://www.open-emr.org/demo/ (accessed May 3, 2023)

[9] Medisafe, Medisafe Pill & Med Reminder. MedisafeApp, https://play.google.com/store/apps/details?id=com.medisafe.android.client&hl=en&gl=US&pli=1 (accessed January 15, 2023)

[10] Getdoc, GetDoc - Search and Appointmen. Jireh Group, https://play.google.com/store/apps/details?id=com.jireh.goseedoc&hl=en&gl=US (accessed February 5, 2023)

[11] J. de V. Mohino, J. B. Higuera, J. R. B. Higuera, & J. A. S. Montalvo, "The application of a new secure software development life cycle (S-SDLC) with agile methodologies", Electronics (Switzerland), vol. 8, no. 11, pp. 1 – 28, 2019.

[12] A. A. Akimov, D. R. Valitov & A. I. Kubryak, "Data Preprocessing for Machine Learning", Scientific Review. Technical Sciences, no. 2, 2022.

[13] R. Meenal, P. A. Michael, D. Pamela & E. Rajasekaran, "Weather prediction using random forest machine learning model", Indonesian Journal of Electrical Engineering and Computer Science, vol. 22, no. 2, pp1208-1215, 2021.

[14] Kaggle. Disease Prediction Using Machine Learning. Kaggle Inc., https://www.kaggle.com/datasets/kaushil268/disease-prediction-using-machine-learning (accessed December 15, 2022)

[15] S.T. Lim, J.Y. Yuan, K.W. Khaw & X. Chew, "Predicting Travel Insurance Purchases in an Insurance Firm through Machine Learning Methods after COVID-19", Journal of Informatics and Web Engineering, vol. 2, no. 2, pp. 43 – 58, 2023.