A logo with black text

Description automatically generated

**NIT6150 Advanced Project**

**Project Evaluation Report**

**HealthCare Chatbot System**

**A hand holding a heart with a pulse line

Description automatically generated**

**Team Leader:** Jitendra Shrestha (s8104215)

**Team Member:** Pranish Acharya (s8100698)

**Client:** Holroyd Private Hospital

**Supervisor:** Fakhra Jabeen

VU Sydney

**Table of Contents**

[1. Overview 1](#_Toc176889717)

[2. Objective 1](#_Toc176889718)

[3. Features 1](#_Toc176889719)

[3.1 Symptoms checker 1](#_Toc176889720)

[3.2 Book an appointment 1](#_Toc176889721)

[3.3 User-Friendly Interface 1](#_Toc176889722)

[3.4 Roles Management 2](#_Toc176889723)

[3.5 User Management 2](#_Toc176889724)

[3.6 Data Privacy 2](#_Toc176889725)

[4. Implementation 2](#_Toc176889726)

[5. Testing 3](#_Toc176889727)

[5.1 Unit Testing 3](#_Toc176889728)

[5.2 UI Testing 3](#_Toc176889729)

[6. Project Completion 4](#_Toc176889730)

[7. Timeline Comparison 4](#_Toc176889731)

[8. Challenges and Problems 5](#_Toc176889732)

[9. Strength and Limitations 6](#_Toc176889733)

[9.1 Strength 6](#_Toc176889734)

[9.2 Limitations 6](#_Toc176889735)

[10. Future Improvements and Lessons Learned 7](#_Toc176889736)

[10.1 Future Improvements 7](#_Toc176889737)

[10.2 Lessons Learned 7](#_Toc176889738)

[11. Potential Privacy Issues 7](#_Toc176889739)

[12. Professional ethics Issues 8](#_Toc176889740)

# 1. Overview

A healthcare chatbot system is an artificial intelligence (AI) technology that functions as a virtual assistant, offering prompt answers to medical questions, symptom evaluations, and tailored guidance to aid users in managing their health. It improves accessibility to healthcare services by understanding and interpreting user input using Natural Language Processing (NLP) to give pertinent information and help. This makes it an invaluable tool for anyone looking for general health advice in between appointments with medical professionals.

# 2. Objective

The objectives of the system are as follows:

* To provide response on health queries
* To diagnose disease based on symptoms provided based on yes/no statement
* To provide way to maintain patient health record

# 3. Features

Healthcare chatbot system has many functions included in it which are explained below:

## 3.1 Symptoms checker

Users can check their ongoing health conditions using chatbot. With the provided data system will provide the home remedies for the related disease. In Case if it is not found it will recommend to the online portals.

## 3.2 Book an appointment

Users can book the appointment with related specialized doctors by signing up. Also, they can check the prescription given by doctor with the follow-up date if required.

## 3.3 User-Friendly Interface

All users can quickly navigate through the application and grasp our website in an efficient manner. From registering, using the log-in feature, signing documents, and exporting the file. These elements are all presented in a logical manner to help our client have a seamless transaction.

## 3.4 Roles Management

System consists of different roles which will limit the user to perform the tasks. One is Admin user who can manage doctors and view doctors. Doctors can add prescription to the booked appointment by the user. Patients can book appointments for the doctors choosing their specialization.

## 3.5 User Management

All users need to login for using the features except only for the chatbot feature

## 3.6 Data Privacy

We have not recorded the chat history when using the chatbot.

# 4. Implementation

For the project implementation process we have worked on all steps sequentially which are explained below.

1. **Initiation**: Specify the goals, parameters, and deliverables of the project. Determine the important parties and compile the healthcare chatbot's first specifications. To describe the project's objectives and vision, draft a project charter.
2. **Planning and Design**: Create a thorough project plan with deadlines, goals, and resource allocation. Wireframes and prototypes should be made to design the chatbot's user interface and experience.
3. **Development**: Construct the chatbot's front end, back end, and components. Django is used to construct the backend while HTML, CSS, and JavaScript are used to implement the user interface. Set up the database using Django ORM and integrate the Rasa Framework for the chatbot.
4. **Testing and Quality Assurance**: To make sure the chatbot functions properly and satisfies quality requirements, do functional, performance, and security testing. Find and address any defects or problems found during testing.
5. **Refinement and Finalization**: Examine and improve the chatbot in light of user input and testing results. Complete the functionality and design to make sure the application satisfies all project criteria and is prepared for deployment.
6. **Deployment and Project Closure**: Install and run the chatbot program in a live setting. Make sure users can access and the application is hosted correctly. Finalize project documentation, carry out a project review, and get input from relevant parties. Ascertain that all deliverables are fulfilled and offer any training or resources that may be required. Archive the project's documentation, then declare it closed.

# 5. Testing

## 5.1 Unit Testing

* We have written and executed unit tests for each function using built-in python module. These tests ensure that individual components of our code are functioning correctly in isolation.
* Each unit test is designed to verify specific functionality and edge cases, helping us identify and fix issues at the function level before integration.

## 5.2 UI Testing

* For user interface testing, we utilized Selenium, a powerful tool for automating web browsers. Selenium allowed us to create automated test scripts that interact with the application's user interface.
* Our UI tests cover various scenarios, including form submissions, navigation, and user interactions, to ensure that the application behaves as expected from an end-user perspective.
* These tests help us verify that the user interface is both functional and user-friendly, catching any issues that may arise in real-world usage.

Together, these testing strategies provide a comprehensive approach to verifying the reliability and quality of our project, from individual functions to the overall user experience.

# 6. Project Completion

To finish the healthcare chatbot project quickly and with minimal data, concentrate on creating a Minimum Viable Product (MVP) that has basic symptom screening, typical health queries, and rapid medical advice. Furthermore, incorporate the chatbot with the personal health management system so that users may schedule doctor appointments. All users can be managed using the user management system, which guarantees smooth communication and efficient platform functionality. This method delivers essential value to users while guaranteeing timely project completion.

# 7. Timeline Comparison

**Index:**

***Green*** *(Represents on track)*

***Red*** *(Represents overspent)*

|  |  |  |  |
| --- | --- | --- | --- |
| S.N. | Project Stages | Time Allocated | Time Spent |
| 1. | **Initiation**   * Kick off meeting * Requirement Gathering * Team Setup | 7 days  1 day  4 days  2 days | 7 days  1 day  4 days  2 days |
| 2. | **Planning and Design**   * Project Plan * **System Architecture Design**   + Wireframes Design   + UI Design * Approval of Design | 6 days  1 day  4 days  2 days  2 days  1 day | 6 days  1 day  4 days  2 days  2 days  1 day |
| 3. | **Development**   * **Backend Development**   + Integration Setup   + API Development   + Database Setup * **Frontend Development**   + Initial Testing   + Chatbot Integration   + UI Development | 16 days  9 days  2 days  5 days  2 days  7 days  2 days  2 days  3 days | **21 days**  14 days  5 days  8 days  1 day  7 days  2 days  2 days  3 days |
| 4. | **Testing and Quality Assurance**   * Unit Testing * System testing * UAT (User Acceptance Testing) | 13 days  4 days  5 days  4 days | 13 days  4 days  5 days  4 days |
| 5. | **Refinement and Finalization**   * Bug Fixing * Feature Refinement * Documentation | 3 days  1 day  1 day  1 day | 3 days  1 day  1 day  1 day |
| 6. | **Deployment and Project Closure**   * Deployment * Training * Project Handover * Post-Deployment Support * Project review and Closure | 3 days  1/2 day  1/2 day  1/2 day  1/2 day  1 day | 3 days  1/2 day  1/2 day  1/2 day  1/2 day  1 day |
|  | **Total Duration** | **43 days** | **48 days** |

***\*Refer to the Appendix for proposed Gantt Chart and Project Schedule***

# 8. Challenges and Problems

During the implementation and testing of the healthcare chatbot and personal health management system, we faced several challenges that required creative solutions.

**Data Limitations**: One significant hurdle was the limited data available for the chatbot, which made it difficult to provide accurate health advice. To address this, we relied on publicly available health data and consulted with medical professionals to validate the chatbot’s responses.

**Integration Issues**: Integration issues arose when trying to connect the chatbot with the existing health management system, particularly around user management and appointment scheduling. We tackled this by conducting thorough API testing and using middleware to ensure smooth communication between the systems.

**Natural Language Processing (NLP) Accuracy**: The accuracy of the chatbot’s NLP was another challenge, especially with the limited training data. To improve this, we created more intents and rules which covers the basic chatbot features.

By addressing these challenges thoughtfully, we were able to deliver a robust and reliable healthcare chatbot system that met users’ needs effectively.

# 9. Strength and Limitations

## 9.1 Strength

**User-Centric Design**: The chatbot is designed to give users quick and easy access to basic health information, making it super convenient for anyone looking for some initial health advice.

**Integration with Health Management System**: By connecting with a health management system for booking appointments and managing user roles, the chatbot offers a one-stop solution that makes managing health easier and more organized.

**Fallback Mechanism**: If the chatbot can’t handle a question, it directs users to healthcare portals. This way, users get accurate help and feel secure knowing there’s always a backup.

## 9.2 Limitations

**Data Dependency**: The chatbot’s effectiveness really depends on the quality and amount of data it has. Without enough good data, it might not always give the best advice.

**NLP Accuracy**: The chatbot might struggle with complex or unclear questions due to its NLP limitations. It might need better training or updates to handle a broader range of user inputs.

**System Integration**: Making sure the chatbot and the health management system work smoothly together can be tricky. Integration issues might affect how well everything works.

# 10. Future Improvements and Lessons Learned

## 10.1 Future Improvements

**Expand Data Sources**: Adding more data sources and keeping the chatbot’s knowledge up to date will help it provide more accurate and relevant health advice.

**Enhance NLP Capabilities**: Investing in better NLP technology and expanding the training data can improve the chatbot’s ability to understand and respond to a wider variety of questions.

**Optimize System Integration**: Working on a smoother integration process and fixing compatibility issues will ensure everything runs more seamlessly.

## 10.2 Lessons Learned

**Importance of User Feedback**: Researching and getting feedback from user is crucial. It helps catch issues and make the chatbot more user-friendly.

**Need for Robust Testing**: Thorough testing is key to catching potential problems before they affect users. This includes checking APIs, performance, and security.

**Scalability Considerations**: Planning for future growth from the start helps avoid performance issues as the user base expands. Keeping the system responsive and efficient is important.

# 11. Potential Privacy Issues

Since personal health information is sensitive, controlling privacy issues is essential in healthcare chatbot projects. Unauthorized access to data, data breaches, regulatory noncompliance, incorrect data retention, insufficient user permission, and insufficient data anonymization are some major hazards. To solve them, put strong access restrictions and encryption in place to stop unwanted access, carry out frequent security assessments to find weaknesses, and make sure that laws like HIPAA and GDPR are being followed by conducting regular audits and training. To safely handle information, establish explicit regulations for the preservation and deletion of data, offer clear permission procedures that educate users about the use of their data, and employ data anonymization techniques for any study or analysis to safeguard user privacy. This may protect sensitive data, uphold regulatory compliance, and foster user confidence by proactively addressing these threats.

# 12. Professional ethics Issues

A healthcare chatbot must consider several ethical considerations, including protecting user privacy by disclosing the chatbot's limitations, maintaining accuracy and dependability through evidence-based information, preventing bias using diverse data, and handling data ownership and usage transparently. The data privacy and confidentiality seem to be the major one so described below clearly.

**Data Privacy and Confidentiality**: For a healthcare chatbot, maintaining data privacy and confidentiality is essential. To comply with laws like HIPAA and GDPR, this entails protecting data with robust encryption, enforcing stringent access restrictions, getting explicit user consent, and performing frequent security assessments. By taking these precautions, users' confidence is preserved, and sensitive data is protected. (Kosinski, 2018)