AI DRIVEN WEB-BASED APPLICATION FOR CHOLERA MANAGEMENT IN KENYA

ALAVI, HAKEEM FAROUK
134775

An Information System Project II documentation submitted to the School of Computing and Engineering Sciences in partial fulfilment of the requirements for the award of the Bachelor's degree in Business Information Technology of Strathmore University

School of Computing and Engineering Sciences
Strathmore University
Nairobi, Kenya.

December 2023

Declaration

I declare that this work has not been previously submitted and approved for the award of a Bachelor’s degree by this or any other University. To the best of my knowledge and belief, the documentation contains no material previously published or written by another person except where due reference is made in the documentation itself.

Student’s signature:

…………………………….………….. [*Signature*]

…………………………….………….. [*Date*]

Approval

The Information System Project II documentation of Hakeem Farouk Alavi was reviewed and approved *(for examination)* by:

Supervisor’s signature:

…………………………….………….. [*Signature*]

…………………………….………….. [*Date*]

Abstract

This research addresses the imperative need for an advanced web-based system dedicated to cholera epidemic control and prevention. Through a systematic approach, it amalgamates Object-Oriented Analysis and Design (OOAD) with the Kanban methodology for methodical development. Employing technologies such as GitHub, JavaScript, R, PHP, HTML, CSS, PhpMyAdmin, and XAMPP, the system's implementation and testing phases are comprehensively detailed.

The machine learning model, a key component, undergoes meticulous implementation through the Knowledge Discovery in Databases (KDD) procedure in R. Problem definition and goal setting, data selection, data cleansing and preprocessing and data transformation among others are outlined. The chosen Logistic Regression 2 model, after thorough evaluation and hyperparameter tuning, is successfully integrated into an intuitive Shiny web application for user-friendly deployment.

System testing, encompassing black box and usability testing, ensures compliance with established requirements. Noteworthy outcomes include successful module redirection, CRUD capabilities, OTP verification, machine learning model deployment, and real-time analytics visualization. Usability testing reveals high learnability, efficiency, and user satisfaction but highlights accessibility challenges.

In conclusion, the cholera web tool proves effective, offering a user-friendly approach to disease risk assessment. Recommendations emphasize reliable internet connectivity and continuous updates for enhanced user experience. Future works envision expanded functionality, real-time data integration, and collaboration with healthcare entities to fortify the application's impact on public health.

Keywords: Cholera, Web-Based System, OOAD, Kanban, Machine Learning, KDD, Logistic Regression, Shiny, Usability Testing, Epidemic Control, Public Health

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to express my deep appreciation to my supervisor and former lecturer, Patrick Shabaya, for his invaluable guidance and support throughout my academic journey. His expertise and dedication have been instrumental in shaping my understanding and fostering my success. I am grateful for his mentorship and belief in my potential.

Chapter 1: Introduction

1.1 Background

Cholera, a waterborne disease caused by the bacterium Vibrio cholerae, was a significant health concern globally, particularly in underdeveloped nations such as Kenya. It was well-established that cholera outbreaks were triggered by the consumption of contaminated food or water, resulting in severe diarrhea and dehydration. According to the World Health Organization, cholera remained a public health concern, causing an estimated 1.3 to 4 million cases and 21,000 to 143,000 deaths worldwide each year (WHO, 2021). Despite efforts focused on improving water and sanitation systems, enforcing hygiene standards, and providing medical assistance, cholera outbreaks persisted, especially in areas with limited access to healthcare and education.

The lack of widely accessible education for all Kenyans represented a significant gap in cholera prevention and management. Many existing educational programs faced challenges in reaching the entire population, particularly those in rural areas. Information on cholera prevention strategies, signs and symptoms, and available treatments remained inadequately disseminated. This information gap hindered individuals from making informed decisions and taking appropriate preventive measures. As reported by the Kenya National Bureau of Statistics in 2019, about 40% of the population lacked access to improved sanitation facilities, emphasizing the need for robust education initiatives (Kenya National Bureau of Statistics, 2019).

Addressing the deficiency in readily available education on cholera was imperative to prevent further fatalities. Since cholera is largely preventable, providing comprehensive and understandable information to all Kenyans could empower people to protect their lives and those of others. By ensuring universal access to educational resources, we aimed to significantly reduce the frequency and impact of cholera epidemics, ultimately enhancing the overall health of the population. (Kwasi-Do Ohene Opuku, 2020)

The study's objective was to develop a user-friendly web-based cholera preventive education program for all Kenyans. The concept was that by delivering in-depth and accessible cholera information, incorporating interactive elements like AI prediction models (Ibrahim, 2021), and utilizing data visualization, individuals would be better equipped to make informed decisions and take preventive measures. The anticipated outcome was that this web-based approach would effectively bridge the knowledge gap on cholera, leading to a substantial reduction in cholera epidemics in Kenya. The project sought to evaluate the effectiveness of the web-based solution in addressing gaps in cholera education and, in turn, contributing to the prevention and management of cholera outbreaks in Kenya.

1.2 Problem Statement

The aspirational goal was the complete eradication of cholera outbreaks and cases in Kenya, ensuring universal access to reliable cholera prevention information, prompt medical care, and robust sanitation systems. Unfortunately, the current reality fell short of this ideal. Kenya continued to grapple with persistent cholera outbreaks, particularly in regions with limited access to healthcare, inadequate sanitation, and insufficient cholera prevention education. The lack of widely available, easily accessible information hampered public awareness, leading to delayed medical care requests and inadequate cholera transmission control. These challenges undermined efforts to effectively prevent and contain cholera outbreaks, resulting in unnecessary suffering and avoidable human fatalities.

Closing the existing gaps in education, accessibility, and information dissemination was crucial to address these issues and move closer to the desired scenario. Establishing a web-based tool that provided comprehensive, easily accessible information on cholera prevention was imperative. This system incorporated a web scraper API for real-time data collection on cholera in Kenya, dynamic data visualization, and an AI prediction model. Ensuring free access to the web application for every Kenyan with a smart device and internet connectivity guaranteed equitable access to life-saving knowledge, empowering individuals to take preventive action against cholera.

By successfully implementing this initiative, we aimed to enhance cholera preventive knowledge, foster behavioural change, and raise public awareness. Consequently, cholera epidemics would occur less frequently, public health would improve, and lives would be saved. Additionally, the research conducted as part of this endeavour contributed to the existing body of knowledge on cholera prevention methods, paving the way for future developments and interventions.

1.3 Aim

The aim was to develop a web-based application powered by AI that provided the people of Kenya with detailed tools and information on cholera prevention.

1.4 Specific Objectives

i. Evaluated the adherence and efficiency of the existing cholera prevention guidelines, including sanitation procedures, hygiene precautions, and health education initiatives.

ii. Reviewed the techniques of education used to maximize the implementation of acquired knowledge into the lives of the people.

iii. Designed a cholera application with an aesthetically pleasing and user-friendly interface that allowed for simple navigation, intuitive interaction, and quick access to key features and information.

iv. Developed a predictive model using artificial intelligence (AI) that utilized symptom alignment to aid in cholera diagnosis in a manner backed by scientific research evidence.

v. Tested the functioning and user experience of the web application by conducting a usability test to gain insight and measure effectiveness.

1.5 Research Questions

i. What were the current cholera prevention guidelines and what was their adherence, as well as how efficient were they?

ii. What were the best techniques of education with the goal of permanently changing regular habits for the better?

iii. What design components and user interface aspects should be used to produce a cholera application interface that is aesthetically pleasing and user-friendly, allowing for simple navigation, intuitive interaction, and smooth access to pertinent data and functionality?

iv. How could Artificial Intelligence be utilized to aid in the prevention and control of cholera in Kenya?

v. How was the web application tested?

1.6 Justification

The research was necessary due to several elements that emphasized the importance and time constraint of resolving the cholera-related issues in Kenya. The project aimed to reduce recurrent cholera outbreaks and enhance public health outcomes by offering a comprehensive web-based solution.

Cholera had been a major public health issue in Kenya, causing fatalities and placing a strain on healthcare systems. The World Health Organization (WHO) reported that there were numerous cholera outbreaks in Kenya in recent years, highlighting the ongoing threat that the disease posed to the populace (WHO, 2023).

Limited information access or the lack of readily available, reliable information was one of the major problems facing the fight against cholera. Many Kenyans found it difficult to get timely, accurate information about cholera prevention, symptoms, and treatment options, especially those who lived in remote areas. Their capacity to defend themselves and stop the disease from spreading was hampered by this knowledge gap (Anis, 2023).

Lack of knowledge and poor hygiene habits also played a role in the persistence of cholera outbreaks. To stop the spread of cholera, it was crucial to practice good hygiene, including handwashing, using clean water, and sanitation. However, research indicated that there was a need to increase public understanding and familiarity with these techniques (Shultz, 2009).

Web-based solutions could break down accessibility barriers and give timely information to a large audience. Studies indicated that digital health treatments, such as web-based apps and AI predictive models, could boost disease prevention tactics and improve health outcomes (Scott Kruse C, 2018). The research aimed to close the information gap and provide Kenyans with accurate and useful facts by utilizing these technologies.

1.7 Scope, Delimitations and Limitations

1.7.1 Scope

To facilitate an increase in cholera prevention, awareness, and knowledge sharing, the proposed study concentrated on tackling the problems associated with the disease in Kenya. The project dealt with the creation and execution of an AI-driven web application that offered thorough details on cholera, including signs and symptoms, precautions to take, and available treatments. With the aid of web scraping, data visualization, and a user-friendly interface, it sought to connect with a variety of Kenyan people while ensuring simple accessibility. The study also included performing usability testing to assess the web application's usability and user experience, enabling incremental adjustments based on user feedback. The study aimed to improve cholera prevention efforts, raise awareness, and equip people with essential information by concentrating on these elements, therefore preventing future outbreaks.

1.7.2 Delimitations

1.7.2.1 Geographical Focus

This study was particularly concerned with Kenya's backdrop and the unique cholera difficulties that the country faced. The conclusions and suggestions might not be immediately relevant to other nations or areas because of their diverse socio-cultural, economic, and healthcare environments.

1.7.2.2 Technology Infrastructure

The suggested web-based solution presupposed that the target population had access to basic technological infrastructure, such as the internet and smartphones. The study made no mention of potential restrictions brought on by insufficient infrastructure in some places, which might have an impact on the reach and efficiency of the online application.

1.7.2.3 User Engagement

The study recognized that factors such as digital literacy, socioeconomic inequality, and cultural preferences had an impact on user engagement and online application uptake. Although attempts were made to create an interface that is user-friendly, the study did not deeply examine user behaviour and adoption patterns.

1.7.2.4 Data Limitations

This study's dataset for cholera symptoms was custom made since there was none that was available online. The predicted difficulties and arduousness involved made it unfeasible to obtain actual data from Kenyan hospitals. As a result, the study's conclusions could be limited by the online dataset's accessibility, accuracy, and completeness. Thus, data regarding cholera prediction was fabricated due to no existing dataset present, so its purpose was to display the functionality of the system and as a proof of concept.

1.7.3 Limitations

1.7.3.1 Sample Size and Representativeness

The study's sample size and representativeness were constrained due to time and resource restrictions. A relatively small number of participants were allowed for the usability testing and user input, and their opinions did not entirely reflect the wide variety of possible users.

1.7.3.2 Data Accuracy

The availability and reliability of publicly available data were essential for the web scraping of cholera-related data for visualization purposes. The accuracy and dependability of the information provided may have been impacted by any restrictions or conflicts in the data sources.

1.7.3.3 Ethical Considerations

Although the study intended to make the online application useful and accessible, it did not address any potential ethical issues that may come from managing personal health information, data privacy, or information security. This study did not address the ethical ramifications of user permission, data storage, or data protection.

1.7.3.4 Time Constraints

The study admitted that time restrictions on the planned research may have affected the breadth and depth of the analysis as well as the feasibility of considering all feasible paths for improvement.

Chapter 2: Literature Review

2.1 Introduction

The literature review section, which offered a thorough comprehension of the most recent knowledge and research relevant to cholera prevention and management, was extremely important to the development of the application. It served as a basis for the application's design, features, and functionality.

Insights were gained into the adherence and effectiveness of current cholera prevention guidelines, education techniques for knowledge implementation, design considerations for user-friendly applications, and the potential of artificial intelligence in cholera treatment by conducting a thorough review of the existing literature. This information directed the development process, ensuring the application incorporated evidence-based methods and tackled the highlighted difficulties and gaps.

The literature research also aided in identifying relevant findings, recommendations, and best practices from earlier studies that were used in the development of the application. It offered a more comprehensive viewpoint on field-tested strategies and interventions that had been successful. Additionally, it made it easier to evaluate user experience research and usability testing done in comparable settings. This data directed the application's testing and evaluation process, ensuring that it satisfied user demands, was simple to use, and offered a satisfying user experience.

In summary, this chapter served as a knowledge base that informed the development process of the application. It helped incorporate evidence-based practices, addressed gaps and challenges, and designed a user-friendly and effective solution for cholera prevention and management in Kenya.

2.2 Review of Specific Objectives

2.2.1 Adherence and Effectiveness of Current Cholera Prevention Measures

The effectiveness of current cholera preventive recommendations was critically assessed in this research review. The objective was to evaluate the degree of compliance, pinpoint the variables that affected compliance, and spot any gaps or limitations in the current guidelines. With the final goal of lowering cholera incidence and enhancing public health outcomes, the review provided evidence for evidence-based interventions and suggestions to improve cholera prevention measures in Kenya.

As a starting point, these prevention guidelines were acknowledged in accordance with a systematic review of the impact of water, sanitation, and hygiene to control cholera. The common prevention guidelines encompassed water quality interventions, hand washing and hygiene interventions (WASH), and the utility of proper storage methods and storage equipment. The majority of interventions fell under water quality, either at the water source or the point of use, with chlorination and filtration being common means. Hand washing and hygiene interventions focused on cleanliness and proper use of storage equipment. In conclusion, while all interventions ideally should be used, a focus on the most impactful, such as hand washing with soap, was recommended, as evidenced by a reduction of about 42-48%, followed by excreta disposal at about 36%, and lastly, water quality interventions at about 17%. (Taylor, 2015)

Compliance to these cholera interventions was investigated, revealing that 90% of households in Lusaka, Zambia, failed to adhere to the cholera interventions, citing various household barriers. These barriers, such as contamination or water leakages, shallow wells, long queues at water points, long distances from water sources, expensive charcoal, expensive chlorine, fear of chlorine, shared toilets, and poor garbage disposal, outweighed enablers like the availability of water, water vessels, knowledge on handwashing, and reception of health information. (Mulenga, 2019)

2.2.2 Review of Knowledge Implementation Education Techniques

The knowledge acquired by a person needed to be maximally utilized and incorporated into their daily lives in the form of a lasting habit. This was especially true when it came to cholera, its prevention, and its management. Prior to commencing, it was crucial to understand what a habit was – a process whereby context prompted action automatically. Habitual behaviour was regulated by an impulsive process and could be elicited with minimum cognitive effort, awareness, or intention. Interventions based on the habit formation model differed from non-habit-based interventions in that they included elements that promoted reliable context-dependent repetition of the target behaviour. Habit formation could realistically be seen as an intervention approach that compromised a broader suite of techniques, marrying context-dependent repetition with strategies that reinforced motivation, boosted action control capacity, opportunity, and skills, facilitated post-initiation repetition, or quickened the learning of associations stemming from repetition.

In conclusion, habit development was a nonlinear process in which the initial repetitions elicited the greatest impact. Therefore, habit formation could be broken down into discrete phases, with the aid of a proposed framework, Lally and Gardner’s framework, that organized habit formation into four interlinked phases. Phase 1 involved making the decision to act, where the intention was formed, and the positive consequences of the action were highlighted, thus enhancing motivation. Phase 2 involved acting on the decision – initiation of intention enactment via the use of action control skills, maintaining behaviour by prioritizing intention over alternatives. Phase 3 entailed repetition where the use of self-regulatory techniques such as planning, setting reminders, and self-monitoring could be utilized. The person should also facilitate the receipt of intrinsic rewards on successful performance of the action. Lastly, phase 4 focused on the manner conducive to the development of cue behaviour associations – pairing action with more frequently encountered cues to quicken habit learning. (Gardner, 2019)

2.2.3 User-Friendly Cholera Application Design

When creating web applications, a user-friendly design, and a seamless user experience (UI/UX) were essential elements. It was crucial to stress the significance of these components in the context of the cholera application. A user-friendly design guaranteed that the software was simple to use, effective, and visually appealing, increasing user adoption. In emergency scenarios where quick access to cholera prevention and treatment resources was crucial, the seamless UI/UX made it simple to access important features and data. The cholera application intended to overcome adoption barriers and provide an effective tool for preventing and resolving cholera cases in Kenya by stressing a user-friendly design and a fluid UI/UX.

A study set out to address the question, "What effect does a poorly designed user interface have on its users' emotions?" in an effort to ascertain whether poorly designed user interfaces (UI) can affect the user's emotions. As a result, a review of prior research in the disciplines of UI/UX, usability, and emotional psychology was carried out to get an overview of the state of the art and to gather the data required to design effective tests for this study. In order to test controllable variables in the subsequent user tests, the authors then intentionally ignored or violated UI/UX best practices and standards in a user interface prototype. In the user tests, participants were given a task to complete using the UI, and while they were doing so, observers noted any difficulties they encountered as well as any emotional reactions they seemed to be experiencing. These observations were later used in a qualitative, semi-structured interview with the participants to get more detail about their experiences.

The authors were able to draw conclusions and provide answers to the study questions after compiling and applying inductive thematic analysis to the test and interview results. Every test taker and interview subject acknowledged experiencing at least one substantial unpleasant emotional response while completing the activity. Numerous diverse emotions, including tension, disappointment, mistrust, and uncertainty, were recorded, but frustration and other rage-related feelings like irritability, impatience, and even anger were reported the most frequently. Most participants stated that they would have left the site before finishing the targeted job if it were a real site. They also demonstrated to create an unfavourable impression of the brand or business in charge of the website, with their negative emotional associations with the user interface passing to the website's owner. The researchers concluded that a bad user interface made people feel bad, probably in the form of rage or irritation. Due to rising bounce rates and a deteriorating public image, this affected not just the user but also the company in charge of the website. (Oscott Fors, 2022)

The evolution of minimalist design, the rise in micro interaction, the popularity of moving pictures, the use of rich colour and sensual typography, and the use of long scrolling and parallax method websites were the main changes in UI/UX trend design aspects during the past few years (Joo, 2017). To sum up, UI/UX design procedures had taken over the creation of digital applications and were now more crucial than ever. These procedures not only made digital programs usable, but they also made them enjoyable to use. The importance of the factor of enjoyability should not be understated because it ensured that a digital application would be used, that it would be used correctly, and that it would ultimately improve the efficiency of the business process it addressed or increase the profitability of the business, depending on the type of digital application being used (Matić, 2021).

2.2.4 AI-Based Predictive Model for Cholera Treatment Development

Artificial intelligence (AI) had the potential to revolutionize healthcare applications, including the treatment and management of cholera. The cholera application sought to increase the precision and efficacy of cholera management by creating a predictive model utilizing AI and symptom alignment. The importance of using AI algorithms to assess symptom patterns and match them with documented cholera cases enabled precise diagnosis and tailored treatment advice. In the end, the cholera application's AI integration aimed to improve patient outcomes, support evidence-based healthcare practices, and open the door for creative AI-driven healthcare solutions.

Artificial intelligence models had made great progress in the medical industry in recent years, transforming disease diagnosis and prognosis, and opening the door for more precise and effective healthcare procedures. Considering this, it was essential to investigate the research that had already been done on illness prediction models driven by AI to comprehend their potential for enhancing early identification and intervention for a variety of medical problems. This cholera prediction model was a logistic regression model because this particular use case was a binary classification; the classification on whether the person had cholera or not.

The relationship between the dependent variable (target), which was categorical data with a nominal or ordinal scale, and the independent variable (predictor), which was categorical data with an interval or ratio scale, was assessed using the predictive model known as logistic regression. To determine the link between the relevant variables, this approach could also be employed in time series modelling. An approach called logistic regression was used to forecast the likelihood of categorical dependent variables. The dependent variable in logistic regression was represented as a binary variable with a value of 1 (yes) or 0 (no). As a function of X, the logistic regression model made predictions. When the dependent variable was dichotomous (binary), the proper regression analysis was done using logistic regression. The mathematical model of logistic regression served as a predictor. When describing data and illuminating the correlation between one dependent binary variable and one or more independent variables at the nominal, ordinal, interval, or ratio level, logistic regression was used.

There were several advantages and disadvantages to logistic regression. These were some advantages of logistic regression: First, logistic regression could demonstrate a meaningful correlation between the independent and dependent variables. Second, you could compare the impact of variables measured at various scales by using logistic regression analysis. Third, the logistic regression model offered information about probability in addition to classification. When using logistic regression, all independent variables had to first have a valid value to produce better results. Second, logistic regression was effective in forecasting multinomial and categorical outcomes. Third, the dataset's variables didn't exhibit multicollinearity.

In accordance with the results of data validation for a study on the use of a Logistic Regression Prediction Model for Cardiovascular Disease, it was discovered that the use of the logistic regression algorithm was effective and efficient in predicting cardiovascular disease, where the accuracy of the prediction results with the algorithm reached 85% with an error rate that tended to be small. The Heart Disease dataset contained fourteen variables. It demonstrated that this approach was appropriate for use in this study as a prediction algorithm (Ciu, 2020). This also pointed us to the direction of the potential utility and effectiveness of this AI model for cholera management.

2.2.5 Usability and Evaluation of the Cholera Application

In order to assess how successfully someone could utilize a product, usability testing involved making direct observations and speaking with potential users of that product. Usability was the ability of a product to be used effectively while tolerating any errors. Usability testing primarily considered user remarks, expressions, and attentiveness throughout tasks, as well as ideas regarding technological advancements. Usability testing for the cholera web-based application utilized the traditional observation method.

Popular interfaces were tested on the basis of common parameters for comparative study. To help illustrate how to use this test, the most typical parameters were presented in the table below:

Table 2. 1 Example of a Usability test's parameters

No.	Parameters for usability evaluation	Cholera web-based application
1	Compatibility with various devices	Responsive on across all devices
2	Search	No
3	Complaint system	Satisfactory
4	Testing techniques	Usability test

5	Feedback form	Yes
6	Security process	Average
7	Network performance	High
8	Security algorithms	None
9	Languages used	JS, PHP, R
10	User friendly interfaces	Yes
11	Back end	PhpMyAdmin
12	Scalability	To be determined
13	Availability	Worldwide

In conclusion, for all types of interfaces, usability was crucial. It aided engineers in creating user interfaces that met user expectations. This review procedure was intended to identify issues with web applications such as security, search, the complaint system, and the online assistance system. It was possible to solve all these issues by tweaking the website's design and testing methods. We presumed that these types of little and large problems could be controlled ahead of schedule if the right testing technique was employed from the first to the last phase of the development process. (Sharma, 2018)

2.3 Research gaps

When it came to specialized solutions for cholera detection, there was a glaring gap despite the fact that web-based solutions had surfaced for a variety of e-health issues, such as cardiovascular diseases. Despite the fact that cholera was prevalent in some areas, there weren't many accessible and user-friendly web tools devoted to identifying and treating this illness. The majority of the material available on cholera was presented in the form of scientific articles and research papers, which made it harder for the common individual to read and comprehend. There was no all-encompassing solution for this problem, and as a result, there was a need for a web-based solution that offered reliable cholera diagnosis in addition to disseminating information in a more palatable and understandable manner.

2.4 Conceptual framework

This conceptual framework provided an improved comprehension of the interconnections and implications of a web-based application for public health interventions in the context of cholera by acting as a guide for data collection and analysis.

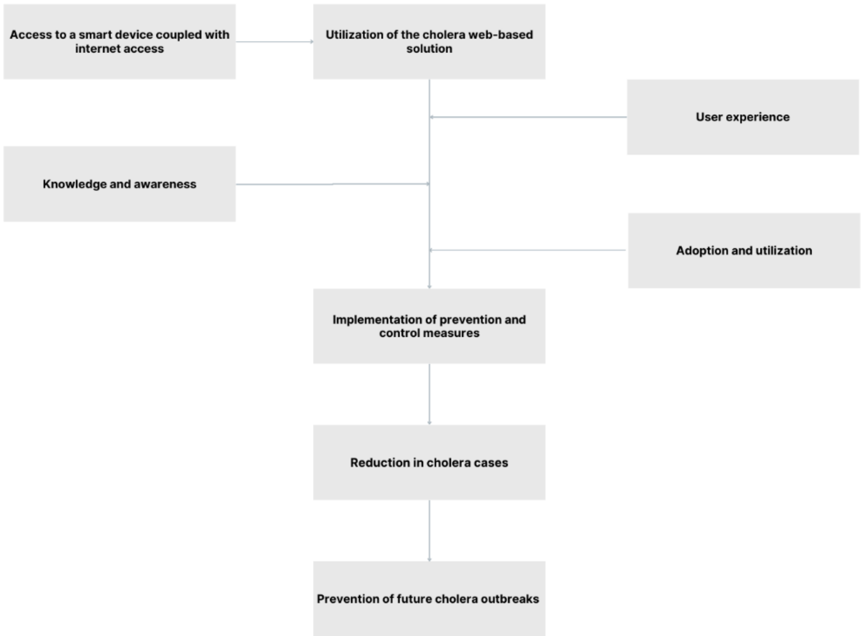


Figure 2. 1 Conceptual Framework

The figure of the conceptual framework above illustrates the variables present and how they relate to each other. The independent variable was utilization of the cholera web-based solution. The control variable was access to a smart device coupled with internet access. The mediator variable was knowledge and awareness while the moderator variables were user experience and adoption and utilization. The dependent variables were implementation of prevention and control measures, reduction in cholera cases, and prevention of future cholera outbreaks.

Utilization of the cholera web-based solution was an independent variable that measured how actively people interacted with and used the web-based tool created specifically for cholera prevention and control. The independent variable represented the factor that was manipulated or observed to examine its influence on the dependent variables.

The availability and capability of people to utilize a smart device (such as a smartphone or tablet) that had internet connectivity were referred to as the control variable - "access to a smart device coupled with internet access." The control variable was utilized to take into consideration any potential effects that technology access and internet connectivity might have on how well people used the cholera web solution. By doing this, you could make sure that any effects on the dependent variables that were seen were caused by the independent variable and not by access to technology.

Knowledge and awareness served as a proxy for people's comprehension and conscientiousness of cholera preventive and control efforts. The mediator variable acted as a connecting factor between the dependent variables and the independent variable (utilization of the web-based solution). It highlighted how using a web-based solution could increase people's knowledge and awareness, which in turn affected how they behaved in terms of cholera prevention.

User adoption and utilization, as well as user experience, served as moderator variables. The term "user experience" described how satisfied people were with the web-based service overall and how simple and effective it was to use. The degree to which people accepted and actively incorporated the web-based service into their daily activities was the difference between adoption and utilization. These moderator variables affected the degree or direction of the association between the dependent variables and the use of the web-based solution. They emphasized that aspects including utilization of prevention and control measures, reduction in cholera cases, and prevention of future outbreaks could be moderated by factors such as user experience and adoption and utilization.

The actual implementation of proposed cholera prevention and control techniques and protocols was referred to as the implementation of prevention and control measures. The term "reduction in cholera cases" referred to a decline in the number of cases that had been officially reported in a given population or location. The preventative steps made to stop cholera epidemics in the future were referred to as "prevention of future outbreaks." These dependent variables were the results that were anticipated to be affected by using the web-based solution, mediated by knowledge and awareness, and controlled by user experience and adoption and utilization.

Chapter 3: System Development Methodology

3.1 Introduction

Object-Oriented Analysis and Design (OOAD) was selected as the System Development Methodology (SDM) method for this web application. Object-oriented analysis and design were software engineering techniques for creating software systems that involved creating object-oriented models that abstracted essential characteristics of the target system and utilized the models to guide the development process (Rumbaugh, 2003). OOAD was favoured for a number of reasons that complemented the project's requirements and objectives. First off, OOAD encouraged modularity and reusability, making the web-based solution easier to maintain and scale. The OOAD abstraction and encapsulation principles ensured a distinct separation of concerns while assisting in understanding the system's behaviour and structure. Additionally, OOAD was well-suited to take advantage of the capabilities of these languages since the project predominantly used programming languages like R and JavaScript, both of which had strong support for object-oriented programming (OOP). Additionally, OOAD offered methods for controlling system complexity and encouraged incremental and iterative development, allowing for ongoing input and improvement throughout the course of a project. OOAD emerged as the best SDM strategy for this project after taking these considerations into account since it provided a comprehensive framework for efficient system analysis, design, and implementation.

3.2 Kanban Method

Agile methodology project management style known as Kanban was initially used in manufacturing and afterwards modified for software development. The word "kanban" is a translation of the Japanese word for "visual signal" or "card". Kanban placed a strong emphasis on visualizing work, minimizing work-in-progress, and maximizing workflow effectiveness. It made use of a visual board with columns designating various task stages, such as "To Do," "In Progress," and "Done." Kanban also enabled individual developers to have a clear perspective of their work and measure their progress without focusing solely on teams. It encouraged a pull-based system, in which work was added to the workflow as capacity permitted, depending on the person's capacity to take on new responsibilities. Kanban was a versatile and adaptive method for properly managing work, enabling single developers to organize their workload, give priority to the most important activities, and maintain a steady and productive development process. (Atlassian, n.d.)

There were five successfully proven principles for Kanban implementation that were: visualizing the work and the workflow that it followed, limiting work-in-progress (WIP) using a virtual Kanban system, managing flow, making management policies explicit and, using models and the scientific method and improving collaboratively (Kirovska, 2015).

3.2.1 Visualise the Work and Workflow

Kanban relied heavily on visualization to ensure that the Work and Workflow were understood clearly. A Kanban board, where tasks were represented as cards and moved across various phases or columns, could be used to accomplish this. For instance, a Kanban board for software development featured columns for "To Do," "In Progress," and "Done," enabling the solo developer or team to see whether tasks were still open, in progress, or finished (Kirovska, 2015).

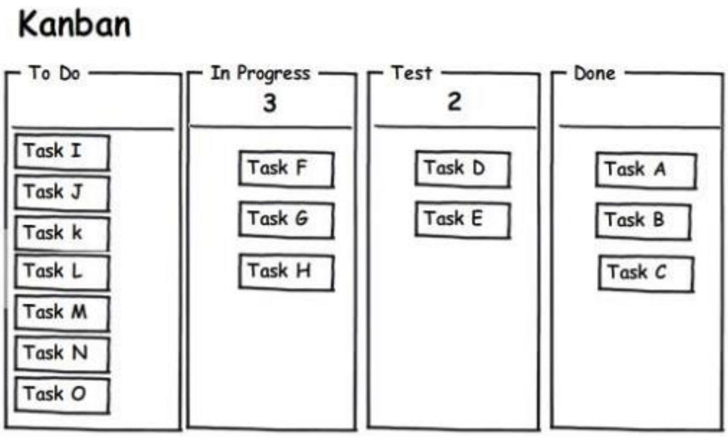


Figure 3. 1 Kanban Board example (Kirovska, 2015)

3.2.2 Limit Work-in-Progress (WIP)

Limiting Work-in-Progress (WIP): Limiting WIP prevented overload and promoted concentration on finishing projects. We could prevent excessive multitasking and ensure that work was finished more quickly by restricting the number of tasks that could be carried out concurrently. The solo developer or team didn't begin new tasks until the number of active tasks in a stage fell below the limit, say, if the WIP limit for that stage was set at three (Kirovska, 2015).

3.2.3 Manage Flow

Monitoring and enhancing the flow of work through the system required constant attention. Maintaining a fluid and consistent flow of work involved locating bottlenecks and dealing with them. For instance, if a developer observed that tasks frequently stalled at a certain level, he might investigate the causes of the holdup and take corrective measures to enhance the flow, such as allocating more resources or streamlining the procedure (Kirovska, 2015).

3.2.4 Explicitly state management policies

Clarity and consistency in decision-making were ensured by outlining management policies explicitly. These procedures specified the order in which tasks were allocated, prioritized, and progressed through the Kanban system. A policy might specify, for example, that tasks should be assigned based on team members' expertise or that urgent customer requests should be given top priority. Making these policies clear guaranteed the awareness of the guidelines and the making of thoughtful decisions (Kirovska, 2015).

3.2.5 Use Models, the Scientific Method and Improve Collaboratively

Kanban promoted an improvement process that was data-driven and collaborative. The solo developer or teams could examine data and find chances for improvement using models and the scientific approach. Teams, for instance, could monitor cycle time—the amount of time it took to perform a task—and use it as the foundation for alterations to their workflow. Team members, stakeholders, and customers all contributed to the improvement process by giving feedback and making recommendations for ways to improve the Kanban system and get better results (Kirovska, 2015).

3.3 Method to be used to Gather the Functional and Non-Functional Requirements

The processes employed to compile the functional and non-functional requirements for the creation of the cholera web-based solution were described in this chapter. Due to the limitations and delimitations previously discussed, feasible methods for assembling the requirements were concentrated on.

The existing literature, research papers, and case studies pertaining to cholera prevention and web-based solutions were the main sources used to compile the functional requirements. A thorough analysis of the literature was done to pinpoint the crucial features and functions that were frequently suggested for and included in systems of a similar nature. The goal was to make sure that the solution included the crucial functional requirements by utilizing the knowledge offered by professionals and researchers in the subject.

An extensive analysis of industry best practices, regulations, and standards in the healthcare and web development domains was done for the non-functional needs. In order to acquire insight into the performance, usability, security, and scalability elements that should be taken into consideration for the solution, pertinent resources were evaluated, and reputable sources were consulted. The non-functional requirements that directed the development process were defined with the aid of this analysis.

To provide clarity and traceability, the requirements were systematically recorded. The functional and non-functional needs were expressed in requirement specification documents in an organized manner. Use case diagrams and entity-relationship diagrams are two examples of visual models and diagrams that were used to describe the requirements graphically and in an intelligible manner.

The emphasis was on utilizing already-existing information and resources to obtain useful functional and non-functional needs. This strategy aimed to guarantee that the solution was in line with accepted best practices and industry guidelines. The requirements acquired acted as the cornerstone for the project's next stages. Based on the defined needs, this methodology allowed the construction of a reliable and efficient web-based cholera solution.

3.4 Design Diagrams to be drawn

Different design diagrams were chosen for the cholera web-based solution project to offer a thorough picture of its major elements. To begin, the Use Case Diagram showed the many user interactions and system functionalities in the first diagram. It provided an overview of the many actors, including users and other systems, as well as how they interacted with the system. In the context of cholera prevention and control, this graphic assisted in identifying the key components and user roles (Visual Paradigm, n.d.).

The Class Diagram, which showed the system's object-oriented design, was the second diagram that was selected. It highlighted the classes, their characteristics, and the connections among them. The system's structure was represented visually in this figure, allowing for a clearer comprehension of the components and how they interacted inside the cholera web-based solution (Visual Paradigm, n.d.).

The Sequence Diagram was used to depict the system's dynamic behaviour. This diagram represented the flow of interactions throughout particular processes or scenarios, as well as the messages that were sent and received between various objects or components. In terms of cholera prevention and control, it demonstrated how various system components cooperated and interacted with one another to accomplish desired results (Visual Paradigm, n.d.).

Additionally, the workflow and business operations within the system were represented by the Activity Diagram. The sequential actions, choices, and concurrent procedures involved in cholera prevention and control were depicted in this diagram. It offered a visual reference for comprehending the sequential steps and actions required for successfully implementing preventive measures (Visual Paradigm, n.d.).

An essential design element for the project to develop a web-based solution for cholera was the Entity-Relationship Diagram (ERD). The ERD offered a visual depiction of the system, much like the Class Diagram, but with a focus on information and relationships. The entities (such as users, diagnoses, and administrative data) were described, along with the relationships between them. The ERD provided an in-depth understanding of the data model and its linkages by providing examples of how data was arranged, stored, and related inside the system. This diagram was essential for cholera management online application's data integrity and effective data management (Visual Paradigm, n.d.).

These five design diagrams: Use Case Diagram, Class Diagram, Sequence Diagram, Activity Diagram, and Entity Relationship Diagram, were used to thoroughly define the project's functionality, structure, behaviour, and workflow. These diagrams were chosen because they offered precise and in-depth insights into the essential components of the proposed web-based solution, laying the groundwork for later development stages.

3.5 List of Development Tools that will be used

To enable efficient and successful implementation during the development phase of the cholera web-based solution project, a selection of carefully chosen development tools were used. These resources were chosen based on how well they worked with the methodology for system development that had been chosen, the goals of the project, and the needs that the literature study had shown.

The Integrated Development Environment (IDE) made use of Visual Studio Code (VS Code). A well-liked and lightweight code editor, VS Code offered a wealth of features like code highlighting, debugging tools, and extensions for other programming languages. It was a good option for independent developers like you due to its adaptability and user-friendly interface.

The project largely used R and JavaScript (JS) as programming languages. Since JavaScript is frequently used in online development, it was perfect for putting the front-end components of the proposed web-based solution into practice. The strong syntax of R, on the other hand, made it ideal for backend development chores like data processing and machine learning integration.

The use of a SQL database management system enabled effective data storage and retrieval. The DBMS chosen for this project was PhpMyAdmin. PhpMyAdmin is a SQL database that is simple and straightforward to use and makes processing data simple. It is suitable for managing the data requirements of the cholera web-based solution.

The project made use of GitHub, a web-based platform for hosting Git repositories, for version control and collaboration. GitHub offered a centralized repository for project file management, facilitating easy collaboration, and guaranteeing accurate version control.

Additionally, several JS and R-specific tools and frameworks were used to accelerate development efforts and improve functionality. The web scraper utilized was also created using a PHP library called Goutte that is a component of Symfony. The analytics of the system were visualized using charts from the Chart.js library. CSS for creating interactive user interfaces, R

Shiny for creating the backend server, and various R libraries for putting machine learning techniques into practice were a few examples.

The cholera web-based solution project benefited from an effective and streamlined development process by utilizing these development tools, which included Visual Studio Code as the IDE, JavaScript, and R as the main programming languages, PhpMyAdmin as the SQL DBMS, and GitHub for version control and collaboration. This ensured the successful realization of the project's goals.

3.6 Methods to be used to test the developed system

The project used black box testing and usability testing, two testing approaches, to assure the quality and dependability of the developed system.

With black box testing, the system was tested without considering its internal organization or implementation specifics. It focused on confirming the functionality and behaviour of the system in light of the given criteria. Different scenarios and inputs were covered by the test cases, enabling thorough testing of the system's capabilities. This testing strategy aided in locating any differences between the system's actual behaviour and its expected behaviour (Beizer, 1995).

To assess the system's usability and user experience, usability testing was carried out. Representative users were observed utilizing the web-based cholera solution, and their comments and recommendations were gathered. This made it easier to pinpoint areas where navigation, user engagement, and general user happiness still needed work (Sharma, 2018).
These two testing approaches were used in the project to improve user experience, find and fix potential problems, and make sure the final web-based solution achieved its goals.

3.7 Domain of Execution

The web served as the primary platform for the cholera web solution's execution. This decision was supported by a number of elements.

The first benefit of a web-based system was accessibility. As long as they had an internet connection, users could access the system from a variety of devices, including desktop and laptop computers, tablets, and smartphones. This adaptability made it possible for users to access the cholera application from various locations and gadgets, improving ease and usability.

A web-based solution could have also made deployment and upkeep simpler. The web server could have been used to centrally integrate updates and upgrades, making it simple for users to access the most recent version of the system without the need for manual updates. This streamlined management and guaranteed that all users had access to the newest features and data.

A web-based solution also fit with the goal of reaching a larger audience. The cholera web-based solution could serve a wider user base, including medical professionals, researchers, and the general public, by taking use of the widespread usage of online browsers. This larger reach could aid in cholera prevention and control by fostering teamwork, spreading knowledge, and raising awareness.

Although the solution was primarily web-based, it is important to note that it could be made to be responsive, adjusting to various screen sizes and device capabilities. This ensured usability and functionality regardless of the device being used and enabled a uniform user experience across different devices.

In conclusion, the cholera solution's selection of a web-based domain of execution offered accessibility, simplicity in implementation and upkeep, and the capacity to reach a larger audience. It was an appropriate and useful option for the project as a result of these factors.

3.8 Proposed Modules and System Architecture

The intended web-based cholera solution was made up of a number of modules, each of which supported a different aspect of the overall system functioning. For the system, the following modules were planned:

3.8.1 User Management Module

Users were able to register for accounts, log in, and access their customized profiles as a result of this module's management of user registration, authentication, and authorization.

3.8.2 Information and Education Module

The causes, signs, treatments, and preventative strategies for cholera were all covered in detail in this module. Additionally, it provided users with educational resources to increase their knowledge and understanding, such as videos, articles, and infographics. It also had a section with the most recent recommendations for cholera prevention and control.

3.8.3 Cholera Reports and Data Analytics Module

This module displayed the latest information regarding cholera cases, and it visualized the data with the aid of charts and graphs to make the information easy to understand for the users.

3.8.4 AI Diagnostic Model Module

An artificial intelligence (AI) diagnostic model trained to anticipate cholera based on symptom alignment was used in this lesson. When users entered their symptoms, the system offered a probability-based diagnosis and suggested treatments.

3.8.5 Administration Module

Access to administrative features was made available to authorized employees using this module, such as system administrators or designated staff. It had a dashboard with interactive charts and graphs that showed key performance indicators (KPIs) and related metrics. Users' interaction with the system, system utilization, and other crucial indicators could all be monitored by administrators.

3.8.6 System Architecture

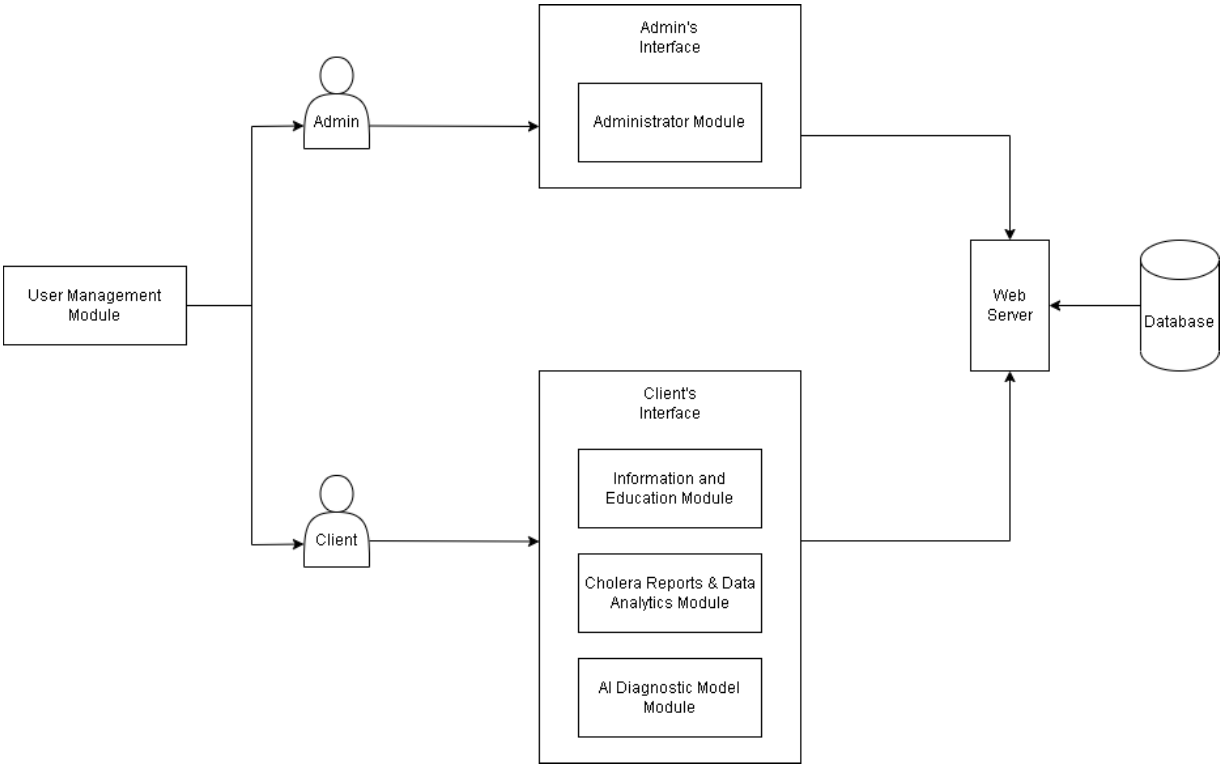


Figure 3. 2 System Architecture

The system architecture diagram identified the key elements of the web-based cholera solution. A reliable web server's placement of the interfaces and modules ensured a safe and usable platform for users. Administrators and clients could interact with the system through the interfaces, specifically the Administrator Interface and Client Interface respectively. The modules played important roles in offering crucial functionalities in this design. Administrators got access to detailed reports and the ability to control system settings by virtue of the Administrator Module.

Moving on to the Client Interface, with the help of instructional resources like videos, articles, and infographics, the Information and Education Module gave users in-depth understanding about cholera, including its causes, symptoms, treatments, and preventative methods. Current data on cholera cases was provided in the Cholera Reports and Data Analytics Module in a visually attractive manner using charts and graphs. An advanced artificial intelligence model that leveraged symptom alignment to provide probabilistic diagnoses and recommended therapies was included in the AI Diagnostic Model Module.

A connected database that acted as a trustworthy repository for user information, system configurations, educational materials, cholera case information, and diagnostic model specifics enabled the seamless integration of all these components. The cholera web-based solution successfully fulfilled the needs of all users in preventing and managing cholera outbreaks thanks to this system architecture, which guaranteed a smooth and efficient operation.

Chapter 4: System Analysis and Design

4.1 Introduction

System diagrams were a straightforward and extremely high-level depiction of a specific system that either already existed or was being constructed (Scala, 2019).The system diagrams that followed this chapter provided a visual representation of the system components and interactions of the cholera management web application. Use Case Diagram, Class Diagram, Sequence Diagram, and Activity Diagram were a few of the diagrams that were created following the OOAD design paradigm.

4.2 System Requirements

The OOAD technique was used because of how extensive the developed system was. According to Evi Septiana Panea (Evi Septiana Panea, 2016), object-oriented analysis and design often identified classes, their relationships to other classes in the issue domain, and the system requirements. Encapsulation, which was essentially the technique of binding properties and methods within the same class, was a component of this paradigm and made it simpler to update a single module without completely upsetting the system at once. The Object-Oriented Analysis Approach, which was compatible with the created system, worked with data rather than methods. The method could be scaled from small to large systems more easily than structured analysis, giving the developer an advantage if a system expansion was needed soon. The functional and non-functional requirements listed below were present in the documented system.

4.2.1 Functional requirements

i. User Registration and Authentication: Users could create an account using their email address, and they could log in using their credentials.

ii. User Data Management: Medical personnel had the ability to manage user records, and those records contained their respective information.

iii. Cholera Diagnosis: Whether the user was likely to have cholera by making a diagnosis.

iv. Reporting and Analytics: Produced cholera case reports and offered analytics to aid in better decision-making.

v. Roles and permissions for users: Assigned appropriate permissions to access and edit particular data by differentiating user roles, such as administrators and users.

4.2.2 Non-Functional requirements

i. Usability: The application adhered to responsive design standards for different screen sizes, and the user interface was simple to use and intuitive for all user types.

ii. Performance: The system loaded rapidly, reacted to user input promptly, and could accommodate many concurrent users without performance degradation.

iii. Security: Only authorized staff had access to critical user information.

iv. Reliability: There was minimum downtime, and the system was accessible and running always. Backups were made often, and data integrity was preserved.

v. Scalability: The program was scalable to handle growing user and data volumes. Both horizontal and vertical scaling were supported by the system architecture.

vi. Compatibility: The application was compatible with a variety of operating systems and web browsers.

4.3 System Analysis Diagrams

4.3.1 Use case Diagram

The interactions between a system and its users were illustrated in the use case diagram, which described the high-level functions of a system. The activities and capacities of each user involved in the system were displayed in the use case diagram below.

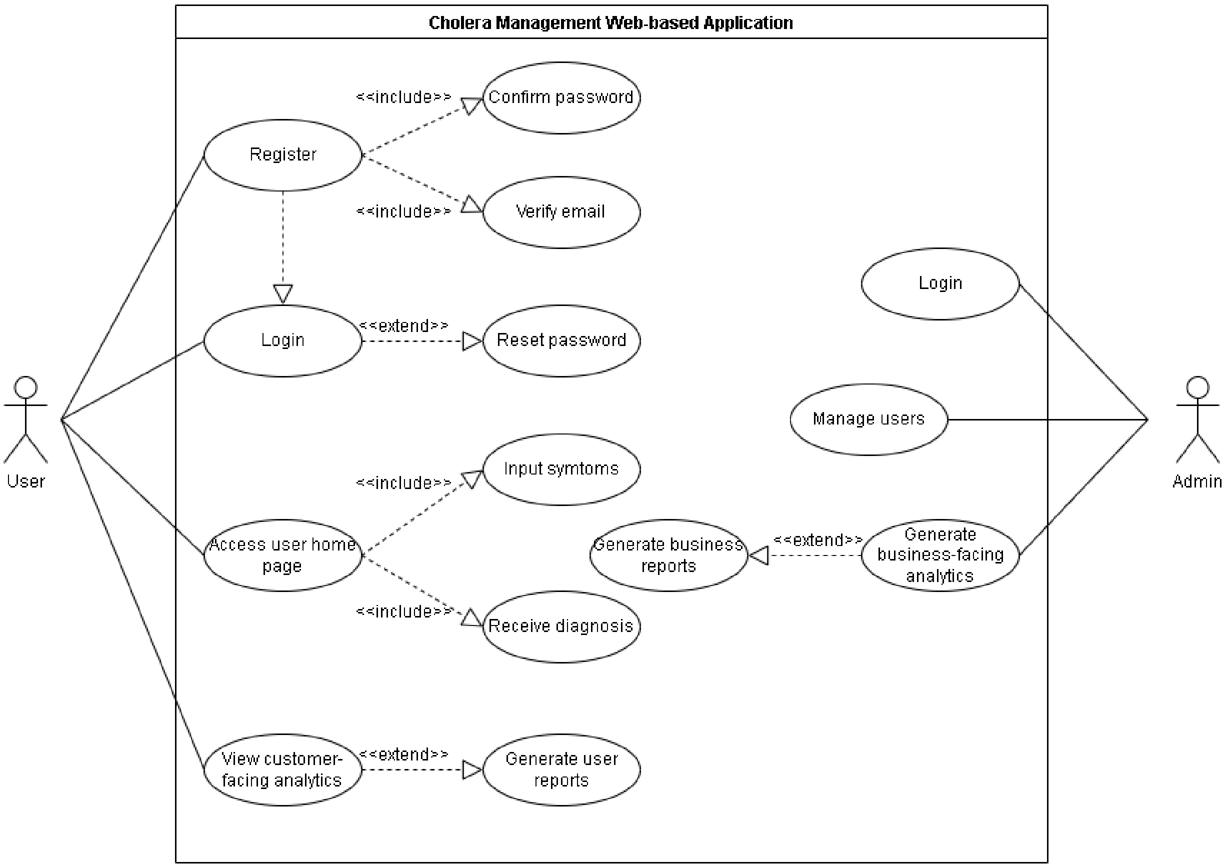


Figure 4. 1 Use Case Diagram

4.3.1.1 Use case Narrative: Initiate Cholera classification

Table 4. 1 Use case Narrative 1

Actor	User and Admin
Precondition	The two actors have to be logged in to the system
Post condition	The user can access and use the cholera application's features
Narration	The user will navigate to the diagnosis page where they can receive their diagnosis
Alternative action	If the user is not logged in, the user should be returned to the login page

4.3.1.2 Use case Narrative: View Cholera diagnosis

Table 4. 2 Use case Narrative 2

Actor	User and Admin
Precondition	The actor should have logged in and the classification should have been initiated first
Post condition	Viewing of classification value
Narration	On the diagnosis page after the initiation of the Cholera diagnosis, the value of the output should be viewed
Alternative action	If the user is not logged in, the user should be returned to the login page and if the diagnosis has not been initiated, the user must do that first

4.3.2 Sequence Diagram

A sequence diagram depicted how things interacted within the framework of a partnership. The objects involved in system development and the order in which messages were sent between them to satisfy a need were shown in the sequence diagram below.

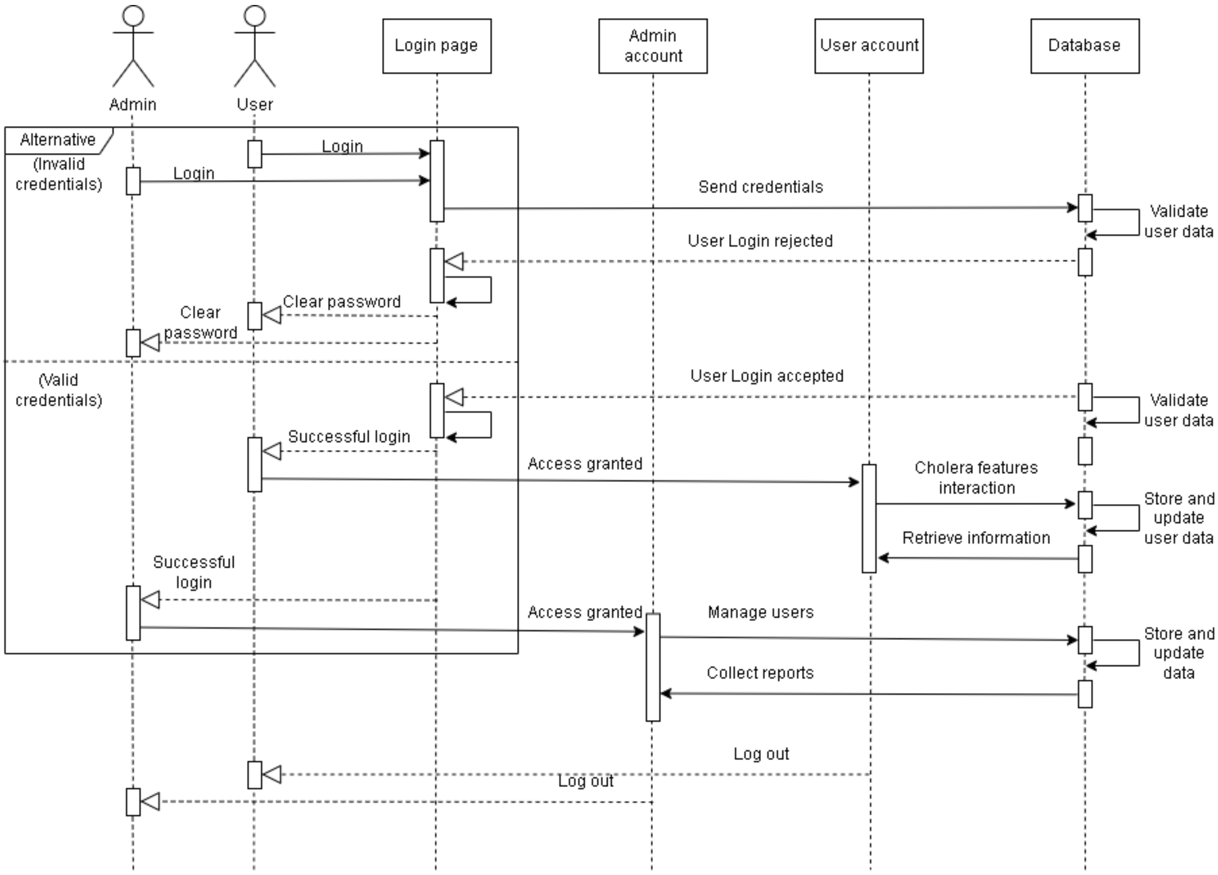


Figure 4. 2 Sequence Diagram

4.3.3 Activity Diagram

In software engineering, an activity diagram was a visual representation that showed the order of steps in a process or workflow. It used symbols like nodes and edges to represent actions, choices, and parallel processing. Activity diagrams gave stakeholders an intuitive representation of intricate processes, making it easier for them to comprehend business processes, software features, and user interactions.

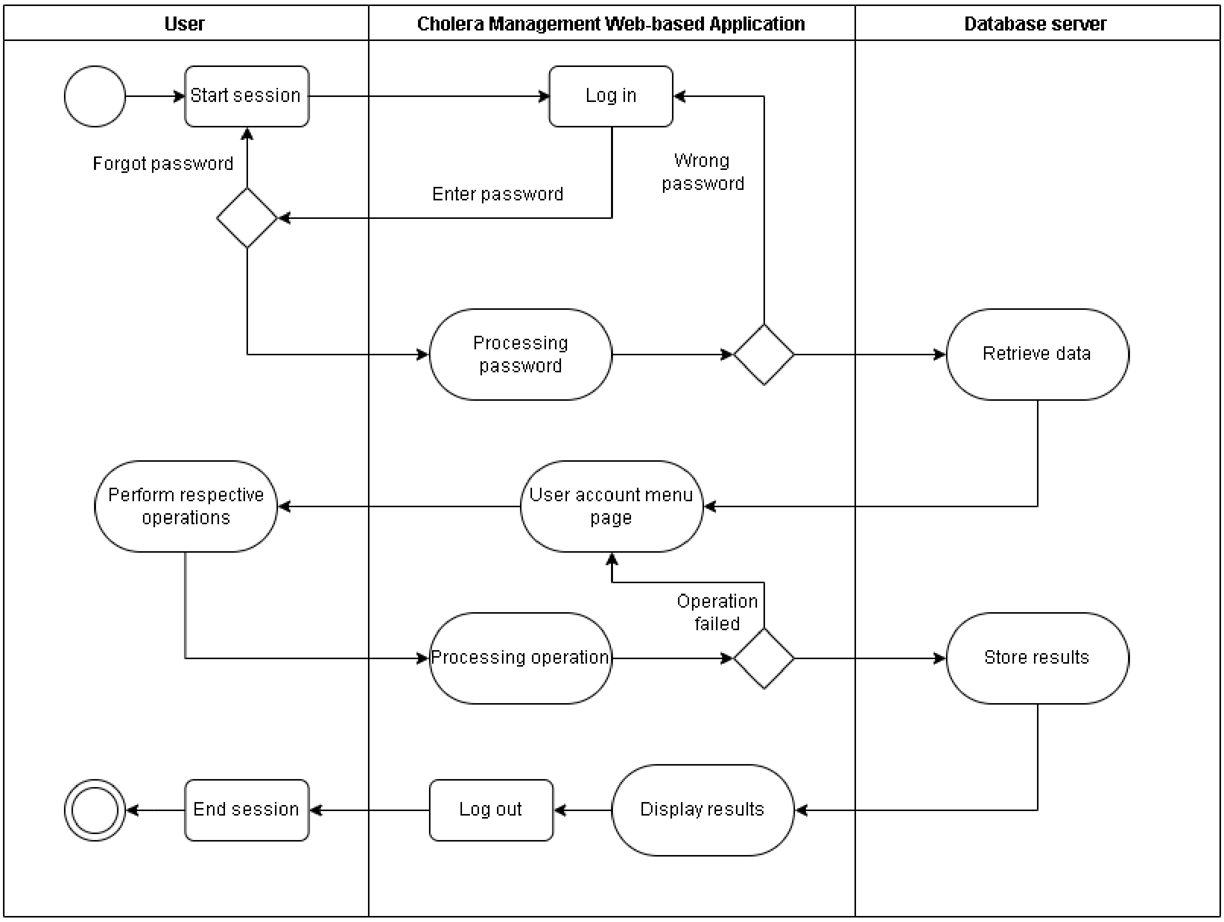


Figure 4. 3 Activity Diagram

4.4 System Design Diagrams

4.4.1 Class Diagram

In software engineering, a class diagram in the Unified Modelling Language (UML) was a form of static structural diagram that depicted the classes, properties, relationships between objects, and actions occurring in the system to explain the structure of a system.

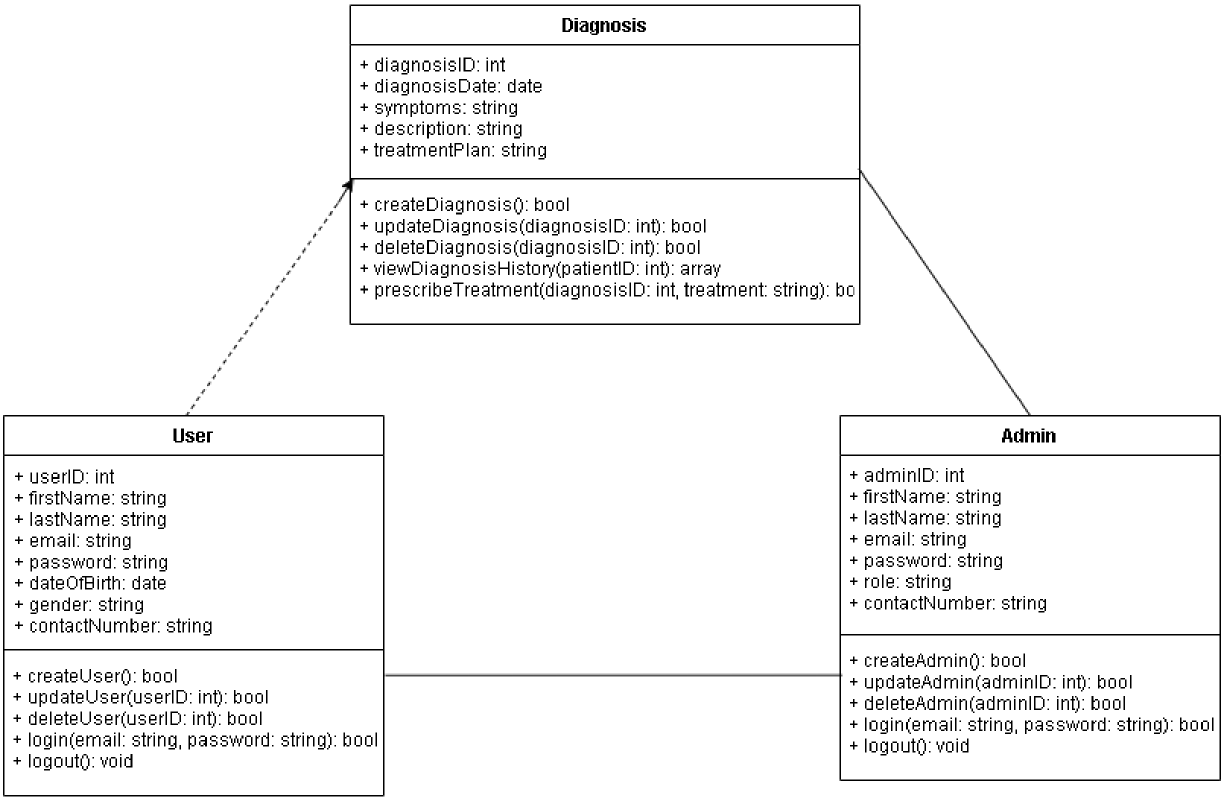


Figure 4. 4 Class Diagram

Chapter 5: System Implementation and Testing

5.1 Introduction

This chapter offers details on how the system was developed. The system's implementation and testing in a local setting will be covered in this chapter. An extensive overview of the developed modules and the satisfaction of both functional and non-functional requirements will be provided.

5.2 Description of the Implementation Environment

5.2.1 Hardware Specifications

The Lenovo IdeaPad S340-14IIL, model number LAPTOP-SIP81LRQ, has an Intel(R) Core (TM) i5-1035G1 CPU running at 1.00GHz, which provides a powerful processing capacity for a range of computing applications thus was the preferred choice for the web application’s development. The system has 8.00 GB of installed RAM, of which 7.78 GB is usable, which enables smooth multitasking and effective resource-intensive application processing.

The product ID for the laptop is 00342-41404-78265-AAOEM. An x64-based processor supports the system type, which is identified as a 64-bit operating system that facilitates the effective handling of huge datasets and software applications.

5.2.2 Software Specifications

The laptop's operating system, Windows 11 Home Single Language, offers a simple and easy-to-use interface for a variety of computing tasks. The version of Windows 11 in question is 22H2, which denotes the most recent version of the operating system that includes the newest features and updates. The operating system build that is installed on the system is Windows 11 version 22621.2428.

XAMPP, a local server housed on the laptop, was utilized to operate the web application. Apache, MySQL, PHP, and Perl were all included in the popular open-source cross-platform web server solution known as XAMPP. It gave developers and testers of dynamic web applications access to an extensive local server environment for development.

Before the potential deployment of web applications to a live server, XAMPP, functioning as a personal development environment, enabled web applications to be tested and debugged. It successfully mimicked the actions of an actual server, allowing server-side programs to run and client requests to be processed in a safe and regulated setting. This configuration made that the application ran in the development environment the way it was supposed to, which made debugging and iterative development easier.

5.3 Machine Learning Model Implementation

Implementing machine learning models requires strict adherence to the methodical and well-structured Knowledge Discovery in Databases (KDD) procedure. There are nine interrelated, iterative processes in this comprehensive approach. These procedures, which include everything from problem definition and goal setting to consolidation, guarantee a careful and comprehensive investigation of the data landscape. The above procedure was utilized throughout the implementation of the Cholera classification model. An overview of the procedure's steps will be provided below.

5.3.1 Problem Definition and Goal Setting

Clearly describing the problem and establishing goals for the creation of the cholera classification model were the main priorities during the first stage of the Knowledge Discovery in Databases process. The necessity to precisely recognize and classify cases of cholera, a serious and sometimes fatal waterborne illness, was highlighted by previous data and research. The main objective was to create a reliable machine learning model that could consistently identify cholera based on certain input feature and subsequently provide an informed guess that the user can utilize. The next step of data selection was made possible by laying the foundation for a clear grasp of the problem landscape and specific objectives.

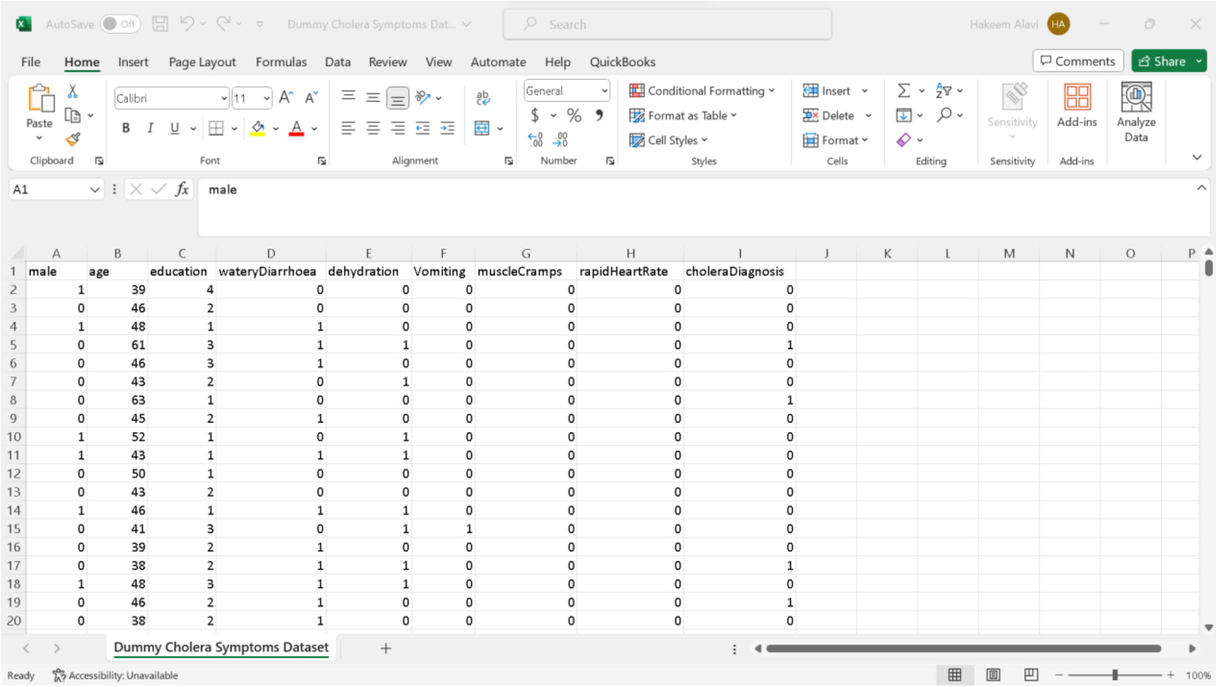
5.3.2 Data Selection

Considering the scarcity of readily available datasets specifically tailored for cholera symptoms, a strategic decision was made to engineer a synthetic dataset in lieu of an authentic cholera symptoms dataset. The absence of comprehensive, structured datasets prompted the creation of a representative dummy dataset simulating symptoms alignment for disease classification. Leveraging an analogous symptoms-aligned dataset obtained from a reputable source on Kaggle, extensive modifications were made to adapt the dataset for cholera classification purposes.

Employing the descriptive statistics toolset, the mean was leveraged as a fundamental measure to discern the prevalence of specific cholera symptoms within the dataset. By calculating the mean across the dataset's binary-encoded columns, each representing the presence or absence of distinct symptoms, a comprehensive overview of the relative frequency of '1's, indicating the occurrence of particular symptoms, was obtained.

The computed mean values were then employed to rank the columns in a descending order, effectively delineating the symptoms with the highest occurrences of '1's to those with the least occurrences. This systematic arrangement facilitated the mapping of cholera symptoms based on their respective frequencies, allowing for a clear delineation of the most common symptoms prevalent within the dataset. The resultant ranking framework enabled a nuanced understanding of the distribution of symptoms.

This pragmatic approach enabled the project to demonstrate a proof of concept, showcasing the potential insights and predictive capabilities achievable if a comprehensive and accurate cholera symptoms dataset were to be assembled in a real-world scenario. By meticulously curating and adapting the existing dataset, the project illustrated the viability and possibilities inherent in the aggregation and consolidation of precise data, thereby laying the groundwork for subsequent stages of data cleansing and preprocessing, and data transformation.



male	age	education	wateryDiarrhoea	dehydration	Vomiting	muscleCramps	rapidHeartRate	choleraDiagnosis
1	39	4	0	0	0	0	0	0
0	46	2	0	0	0	0	0	0
1	48	1	1	0	0	0	0	0
0	61	3	1	1	0	0	0	1
0	46	3	1	0	0	0	0	0
0	43	2	0	1	0	0	0	0
0	63	1	0	0	0	0	0	1
0	45	2	1	0	0	0	0	0
1	52	1	0	1	0	0	0	0
1	43	1	1	1	0	0	0	0
0	50	1	0	0	0	0	0	0
0	43	2	0	0	0	0	0	0
1	46	1	1	1	0	0	0	0
0	41	3	0	1	1	0	0	0
0	39	2	1	0	0	0	0	0
0	38	2	1	1	0	0	0	1
1	48	3	1	1	0	0	0	0
0	46	2	1	0	0	0	0	1
0	38	2	1	0	0	0	0	0

Figure 5. 1 Cholera Symptoms Dataset

5.3.3 Data Cleansing and Preprocessing

Loading the dataset into the R environment was the first step in the data cleansing and preprocessing phase of developing the cholera classification model. The loading process made it easier to create a uniform, organized dataset, which laid the groundwork for the subsequent steps of preprocessing and data cleansing.

```r
# Importing Dataset ----

library(readr)
cholera <- read_csv("data/cholera.csv")
View(cholera)
```

Figure 5. 2 Data Loading

To gain a deeper understanding of the dataset, several descriptive statistics were produced during the preliminary stages of exploratory data analysis. The dimensions of the dataset, the types of data, the evaluation of frequency distributions, and the mode extraction for categorical variables were all included in these statistics. Furthermore, the Multivariate Imputation by Chained Equations (MICE) technique was adopted in response to the dataset's missing data, which was categorized as Missing Not at Random (MNAR). With the help of this complex imputation technique, missing data is iteratively imputed in a chained fashion while considering variable correlations for increased accuracy. After imputation, metrics including variance, standard deviation, kurtosis, skewness, correlation, and covariance were calculated in the second stage of descriptive statistics. Robust data preprocessing and feature engineering procedures were made easier by these metrics, which offered a thorough insight of the dataset's variability, distribution shape, and interdependence among various characteristics.

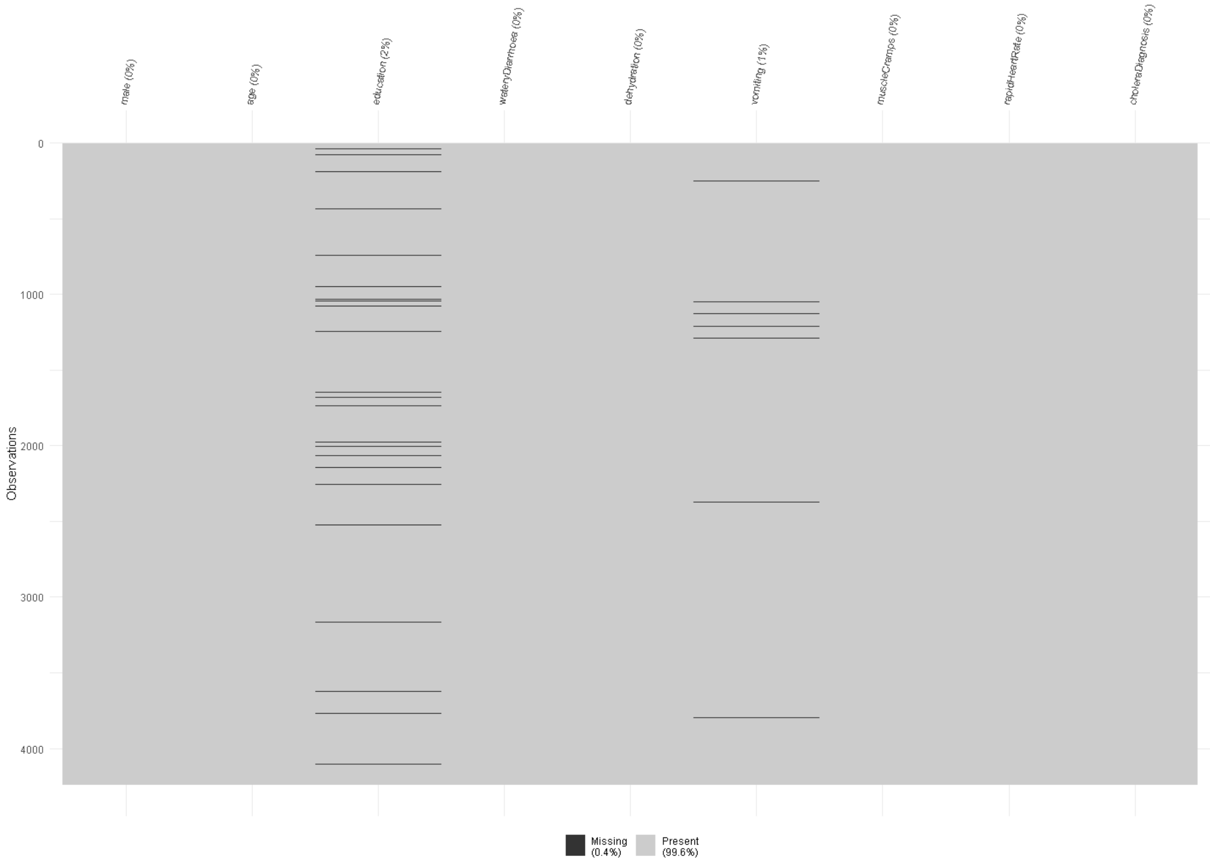


Figure 5. 3 Pre-Imputation Graph

```r
# Create an empty predictor matrix

predictorMatrix <- matrix(0, ncol = ncol(cholera), nrow = ncol(cholera),
                          dimnames = list(names(cholera), names(cholera)))

# Specify the imputation methods for the 'education' and 'vomiting' columns

predictorMatrix['education', ] <- "polr"
predictorMatrix['vomiting', ] <- "logreg"

# Run mice with the specified imputation methods

cholera_mice <- mice(cholera, m = 11, method = 'pmm', seed = 7, predictorMatrix =

View(cholera_mice)
```

Figure 5. 4 Data Imputation

Figure 5. 5 Post Imputation Graph

5.3.4 Data Transformation

The intention of the data transformation step was to improve the machine learning model's performance by making changes to the dataset. Undersampling was one of the main strategies utilized, which lowers the number of members of the dominant class in order to balance the distribution of classes. Undersampling is the process of removing examples at random from the overrepresented class in order to bring it into balance with the minority class. By preventing the model's training from being biased toward the majority class, this procedure helps to lessen the problems caused by unbalanced data and improves classification accuracy.

```
# Data Transformation ----

class_distribution <- table(cholera_imputed$choleraDiagnosis)
print(class_distribution)

## Undersampling ----
# Convert 'choleraDiagnosis' to a factor

cholera_imputed$choleraDiagnosis <- as.factor(cholera_imputed$choleraDiagnosis)

# Identify the positive and negative class samples

positive_samples <- cholera_imputed[cholera_imputed$choleraDiagnosis == 1, ]
negative_samples <- cholera_imputed[cholera_imputed$choleraDiagnosis == 0, ]

# Undersample the majority class (in this case, the negative class)

negative_samples_undersampled <- negative_samples[sample(nrow(negative_samples), nrow(positive_samples)), ]

# Combine the undersampled negative class with the positive class

cholera_undersampled <- rbind(negative_samples_undersampled, positive_samples)

# Check the class distribution after applying undersampling

table(cholera_undersampled$choleraDiagnosis)

# Conclusion: The class distribution is now balanced after performing undersampling
# This balance in the number of instances for each class will help mitigate the
# effects of class imbalance and ensure that your logistic regression model can
# learn from both classes effectively, leading to more accurate predictions and improved model performance.
```

Figure 5. 6 Data Undersampling

5.3.5 Selection of the Data Mining Task

The KDD process's intended objective was the main factor considered while choosing the data mining task. The binary classification data mining problem was correctly chosen, given the goal of classifying individuals as either cholera patients or not. The goal of this classification exercise was to identify, from the particular symptoms that were present in each case, whether or not the individual was suffering from cholera. The following stages of the procedure were coordinated to create a model that could precisely predict the presence or absence of cholera based on the input symptom data by framing the data mining task as binary classification.

5.3.6 Selection of the Data Mining Algorithm

Several performance measures were assessed while comparing the outcomes of three models: Random Forests, Logistic Regression 1, and Logistic Regression 2. The first model, Random Forests, proved to be effective in correctly recognizing true positive cases, as evidenced by its accuracy of 0.6646, Kappa value of 0.3292, and sensitivity of 0.7205. Its specificity of 0.6087, however, indicated a comparatively poorer capacity to accurately identify true negative cases.

Now let's look at Logistic Regression 1, the second model, which showed a Kappa value of 0.3168 and an accuracy of 0.6584. Though slightly less accurate than the first model, it demonstrated a balanced performance in capturing both positive and negative cases, with a sensitivity of 0.6770 and a specificity of 0.6398.

Finally, "glmnet," or Logistic Regression 2, produced an accuracy of 0.6646 and a Kappa value of 0.3292. Its measured values for specificity and sensitivity were 0.6522 and 0.6770, respectively. This model demonstrated a notable capacity to identify true negatives, suggesting that it can reliably categorize persons who are not cholera patients.

After a thorough examination of these results, it was discovered that although the three models performed similarly overall, Model 3 (also known as "glmnet," or Logistic Regression 2) stood out because of its somewhat higher specificity than the other models. This feature suggested that it could identify genuine negatives more correctly, which would make it a more dependable method of identifying those who do not have cholera. Consequently, Model 3 was suggested for more examination and implementation, guaranteeing a more accurate categorization, and diminishing the possibility of false negatives in the cholera diagnosis procedure.

```r
## Logistic Regression 2 ----
# We can use "glmnet"

set.seed(7)
cholera_caret_model_logistic_two <- train(
  choleraDiagnosis ~ .,
  data = cholera_train,
  method = "glmnet",
  family = "binomial",
  metric = "Accuracy",
  preProcess = c("center", "scale"),
  trControl = trainControl(method = "cv", number = 5, verboseIter = TRUE),
  maxit = 1000
)

# Check for multicollinearity
findLinearCombos(cholera_train[, -which(names(cholera_train) %in% "choleraDiagnosis")])

print(cholera_caret_model_logistic_two)

### Calculating Metrics ----
# Make predictions

cholera_predictions_lr_two <- predict(cholera_caret_model_logistic_two,
                                      cholera_test)

# Display the model's evaluation metrics

cholera_confusion_matrix_lr_two <-
  caret::confusionMatrix(cholera_predictions_lr_two,
                         cholera_test$choleraDiagnosis)
print(cholera_confusion_matrix_lr_two)
```

Figure 5. 7 Selected Algorithm

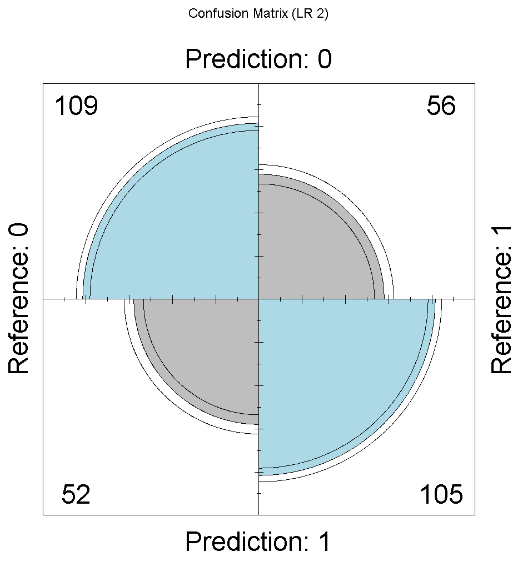


Figure 5. 8 Confusion Matrix

5.3.7 Utilization of the Data Mining Algorithm

Optimizing the performance of the selected Logistic Regression 2 (glmnet) model through hyperparameter adjustment was a laborious procedure. The predictive power of the model was improved by carefully adjusting several parameters, including alpha and lambda, within predetermined limits. Even after all these painstaking modifications and many runs, the result stayed the same—the same output was consistently produced. This result implies that the model's original configuration was already close to its ideal state and that considerable improvement could not be achieved by adjusting hyperparameters alone. While the approach did not produce any significant improvements, it did reveal important information about the robustness of the model and showed that the baseline configuration was already suitable for the particular classification task.

```
# Hyperparameter Tuning ----

library(caret)

# Create a grid for tuning

grid <- expand.grid(alpha = 0:1, lambda = seq(0.001, 0.1, by = 0.001))

# Tune the model

cholera_caret_model_logistic_two_tuned <- train(
  choleraDiagnosis ~ .,
  data = cholera_train,
  method = "glmnet",
  family = "binomial",
  metric = "Accuracy",
  preProcess = c("center", "scale"),
  trControl = trainControl(method = "cv", number = 5, verboseIter = TRUE),
  maxit = 1000,
  tuneGrid = grid
)

print(cholera_caret_model_logistic_two_tuned)

# Make predictions
cholera_predictions_lr_two_tuned <- predict(cholera_caret_model_logistic_two_tuned,
                                            cholera_test)

# Display the model's evaluation metrics
cholera_confusion_matrix_lr_two_tuned <- caret::confusionMatrix(cholera_predictions_lr_two_tuned,
                                                                 cholera_test$choleraDiagnosis)
print(cholera_confusion_matrix_lr_two_tuned)
```

Figure 5. 9 Hyperparameter Tuning

5.3.8 Interpretation and Evaluation

It was crucial to recognize that the model's classification might not have been totally accurate, given how important people's health and wellbeing are. It was therefore seen as a tool to help determine whether or not a person needed medical assistance by providing an informed guess of their cholera status. Moreover, the output of the model may have been applied to subjectively assess the degree of the patient's illness. It was important to stress, nevertheless, that the model should not be used in place of a thorough diagnosis made by a licensed physician or other healthcare provider; rather, it should only be used as an additional tool. The aim of the study was to offer constructive perspectives that would have encouraged people to promptly seek appropriate medical advice and treatment, leading to better health outcomes and general well-being.

5.3.9 Consolidation

Through the creation of a Shiny online application, the model was incorporated into an intuitive user interface to expedite the deployment process. End customers now have extremely easy access to and convenience with the deployment thanks to the shinyapps.io server hosting the application. There were various reasons why this was the best option. First off, the Shiny framework made it possible to create an engaging and eye-catching user interface that made interacting with the model's predictions simple. The ease with which users could enter their data and quickly receive the model's output improved the user experience as a whole. Furthermore, the shinyapps.io server deployment offered a smooth and hassle-free hosting option, doing away with the necessity for time-consuming server setup or maintenance. This strategy made technical complexity and operational overhead far lower, which made deployment go more smoothly. In addition, the shinyapps.io platform provided strong data privacy and security protections, guaranteeing the privacy and confidentiality of sensitive user data. Overall, the model's deployment as a Shiny app on the shinyapps.io server offered a reliable, efficient, and safe approach that made it widely accessible and useful for a variety of users.

5.4 System Testing

This section looks into and reports on the system's capacity to meet the functional and non-functional requirements that were established during the system's development phase. Examining whether the system behaved appropriately when given the correct inputs was one of the testing's goals. To make sure everything was working as it should, the system was also tried with incorrect inputs. The developed technology was put to the test via black box testing.

5.4.1 Black Box Testing

Software testing known as "black box" testing is totally dependent on software requirements and specifications and primarily concentrates on the input and output of software programs. To identify potential issues, the real program execution within the system was analysed. It was also taken into consideration how the system behaved in response to various inputs. To examine the behaviour of the system, some inputs were carefully selected as test data (Ashtari, 2022).
In order to make sure that the mobile application serves its intended purpose for users who are not sufficiently knowledgeable about the inner workings of the system and can verify whether the desired output was created, black box testing was employed. Listed below are the testing results of the black box testing.

Table 5. 1 Black Box Test Results

Test ID	Description	Test Data	Outcome	Status	Evidence
BB01	User redirected to respective module upon logging in based on their assigned system roles	abdulhakeemalavi49@gmail.com and 12345 (Admin), hakeem.alavi@strathmore.edu and 12345 (User)	Admin redirected to the admin dashboard page and user redirected to the user home page	PASS	64
BB02	Admin has full CRUD capabilities over the users	James Smith – Initial name of one of the users	Changed to James Sully	PASS	65
BB03	Verification OTP system works successfully	xiiicapitan@gmail.com – Email used for user sign up	6-digit OTP received in stated email	PASS	65

BB04	Users can successfully logout of their system accounts	abdulhakeemalavi49@gmail.com and 12345 (Admin), hakeem.alavi@strahtmore.edu and 12345 (User)	Successfully redirected to the system landing page	PASS	65
BB05	Cholera classification model deploys and returns an outcome upon user's symptoms submission	Default age (25) as well as default choices (NO) in symptom fields used	A "NO" classification is produced	PASS	66
BB06	User analytics are updated in real time and visualized as charts	Model feedback captured post-classifications and subsequently visualized in an account with the following credentials: hakeem.alavi@strahtmore.edu and 12345 (User)	Visualized classification appended from 1 to 2	PASS	66
BB07	Respective user can access their profile page and alter their personal information	The user John Doe whose login credentials are: xiiicapitan@gmail.com and 12345 (User)	Name successfully changed to John Joe	PASS	67
BB08	User can review the system by submitting a rating	The user John Joe whose login credentials are: xiiicapitan@gmail.com and 12345 (User)	A rating of 4 stars provided and the 'my rating' card updated to 4/5	PASS	67

BB09	User can access, take, and receive corresponding score on the hygiene and sanitation quiz	The user John Joe whose login credentials are: xiiicapitan@gmail.com and 12345 (User), and the 3rd choice of each question was selected as the answer	The score of 8/15 was awarded to the respective user	PASS	68
BB10	Respective users can access their analytics pages and download their reports as PDF files	abdulhakeemalavi49@gmail.com and 12345 (Admin), hakeem.alavi@strathmore.edu and 12345 (User)	The reports containing the analytics as of that specific time were successfully downloaded	PASS	68
BB11	The web scraper used in the news page successfully collects current cholera data and displays it from the WHO website	No testing data required as it can be accessed directly since it is in the landing module	The page took approximately 7 seconds to load before it displayed the relevant information	PASS	69

5.4.2 Usability Testing

A systematic assessment method called usability testing determines how simple and efficient a product, like a website or application, is for consumers to interact with. To improve the overall user experience, designers can find and fix usability flaws by watching actual users complete certain tasks.

A 5-point Likert scale for the Scale field has been used to capture the evaluation, with 1 being the lowest score and 5 representing the highest. Depending on their experience, users can rank every parameter. The table offers a methodical way to collect insightful data about how well your cholera web application is used.

Table 5. 2 Usability Test Results

Usability Parameter	**Description**	**Scale**	**Comments**
Learnability	How easy is it for users to learn how to use the cholera web application?	5	The cholera web application utilizes good industry standard UI/UX principles thus dampening the user's learning curve
Efficiency	How quickly can users perform key tasks using the cholera web application?	4	With the exception of the web scraper, all features perform extremely well without noticeable delays
Navigation	How intuitive and straightforward is the navigation within the application?	5	The top and side navigation bars are omnipresent and tuck away neatly when inactive to maximise the user's screen real estate

Error prevention and recovery	How well does the application prevent errors, and if they occur, how easily can users recover?	3	The system is able to anticipate, detect and handle errors most of the times like providing the user with user friendly error messages that enables the user to swiftly rectify or deal with the error
Satisfaction	To what extent are users satisfied with the overall experience of using the cholera web application?	4	The cholera web application delivers what was promised to the users but still has a large margin for improvement
Accessibility	How accessible is the application to users with disabilities?	1	Unfortunately, the current state of the web application does not cater to people with special needs
Consistency	Is the design and functionality consistent across different sections of the cholera web application?	5	The design and functionality of the system remains consistent therefore retaining a sense of familiarity across all sectors
Feedback	Does the application provide informative feedback to users about their actions and system status?	5	The system always provides the relevant information to the

			user after an action is performed
Aesthetics	How visually appealing and aesthetically pleasing is the cholera web application?	5	The web application has a good contrast of colours, and they all play a role in communicating the importance of an action to the user
Task success rate	What percentage of tasks were successfully completed by users during the test?	5	All users were able to complete all tasks successfully, but it is important to note that the pool of users was small thus a small test size

Chapter 6: Conclusion, Recommendations and Future Works

6.1 Conclusion

In conclusion, the cholera web tool has effectively fulfilled its main goal of giving people an easy-to-use way to determine their risk of contracting the disease. The application enables users to make well-informed decisions on their health by incorporating multiple characteristics, such as the cholera classification model. It is a useful instrument for encouraging early intervention and raising public health awareness.

6.2 Recommendations

It is advised that users use dependable internet-connected devices to use the cholera online application in order to maximize its usability and accessibility. To guarantee a smooth and efficient user experience, regular updates and enhancements based on user feedback should be given top priority. The accuracy and functionality of the application can be improved over time through cooperation with organizations and health professionals.

6.3 Future Works

Future plans for the cholera web-based application may entail broadening its functionality to serve a wider user base by adding more health-related features. Real-time data integration can be facilitated by cooperation with healthcare facilities, increasing the responsiveness and usefulness of the application. The functionality and engagement of the application will be further improved with the addition of personalized features like notifications and updates for user accounts.

The investigation of collaborations with healthcare facilities and community initiatives will be essential to maintaining the application's long-term beneficial effects on public health as part of continuing initiatives. The cholera web-based application is well-positioned to undergo further advancement and enhancement, given the dynamic nature of both technology and healthcare, which demands constant adaptation and innovation.

References

Anis, H. O. (2023). Multicountry cholera outbreak alert in Kenya: Current efforts and recommendations. *International Journal of Surgery*, 555-557.

Ashtari, H. (2022). *Black Box vs. White Box Testing: Understanding 3 Key Differences.*

Atlassian. (n.d.). *Kanban - A brief introduction*. Retrieved from Atlassian: https://www.atlassian.com/agile/kanban

Beizer, B. (1995). *Black-box testing: techniques for functional testing of software and systems.* John Wiley & Sons, Inc.

Ciu, T. a. (2020). Logistic regression prediction model for cardiovascular disease. *International Journal of New Media Technology*, 33-38.

Evi Septiana Panea, R. S. (2016). Capability maturity model integration (CMMI) for optimizing object-oriented analysis and design (OOAD). *Procedia Computer Science,* 40-48.

Gardner, B. a. (2019). *Habit formation and behavior change.* Oxford research encyclopedia of psychology.

Ibrahim, I. A. (2021). The role of machine learning for diagnosing diseases. *Journal of Applied Science and Technology Trends,* 10-19.

Joo, H. (2017). A study on understanding of UI and UX, and understanding of design according to user interface change. *International Journal of Applied Engineering Research,* 9931-9935.

Kenya National Bureau of Statistics. (2019). *Kenya Integrated Household Budget Survey 2015-2016*. Retrieved from Kenya National Data Archive (KeNADA): https://statistics.knbs.or.ke/nada/index.php/catalog/13

Kirovska, N. a. (2015). Usage of Kanban methodology at software development teams. *Journal of applied economics and business*, 25-34.

Kwasi-Do Ohene Opuku, N. C. (2020). *The role of control measures and the environment in the transmission dynamics of cholera.* Abstract and Applied Analysis.

Matić, F. (2021). *Best UI/UX practices in the world of modern IT business applications.* Zagreb: Doctoral dissertation, University of Zagreb. Faculty of Economics and Business. Department of Informatics.

Mulenga, B. (2019). *Adherence to cholera interventions: associated factors, enablers and barriers in Chipata and George compounds of Lusaka, Zambia.* Lusaka: University of Zambia.

Oscott Fors, E. a. (2022). *How User Interfaces influence users' emotions: A qualitative study of users' emotional reactions whileusing a booking system breaking UI/UX best practices.* JTH, Department of Computer Science and Informatics.

Rumbaugh, J. (2003). Object-oriented analysis and design (OOAD). *Encyclopedia of Computer Science*, 1275-1279.

Scala, R. D. (2019). *System Diagram.*

Scott Kruse C, K. P. (2018). Evaluating barriers to adopting telemedicine worldwide: A systematic review. *Journal of Telemedicine and Telecare*, 4-12.

Sharma, R. K. (2018). Comparative Study for Evaluating the Usability of Web Based Applications. *4th International Conference on Computing Sciences (ICCS),* 94-97.

Shultz, A. J. (2009). Cholera outbreak in Kenyan refugee camp: risk factors for illness and importance of sanitation. *The American journal of tropical medicine and hygiene*, 640-645.

Taylor, D. L. (2015). The impact of water, sanitation and hygiene interventions to control cholera: a systematic review. *PLoS one*.

Visual Paradigm. (n.d.). *What is Activity Diagram?* Retrieved from Visual Paradigm: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/

Visual Paradigm. (n.d.). *What is Class Diagram?* Retrieved from Visual Paradigm: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/

Visual Paradigm. (n.d.). *What is Entity Relationship Diagram (ERD)?* Retrieved from Visual Paradigm: https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/

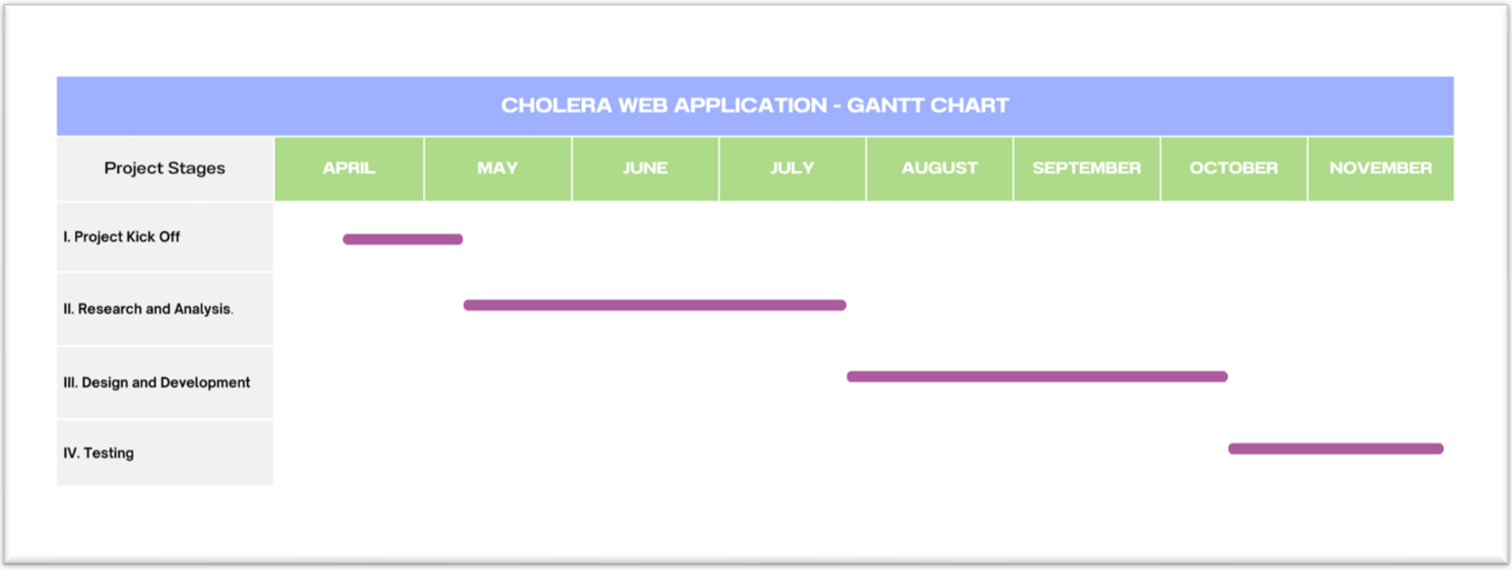
Visual Paradigm. (n.d.). *What is Sequence Diagram?* Retrieved from Visual Paradigm: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/

Visual Paradigm. (n.d.). *What is Use Case Diagram?* Retrieved from Visual Paradigm: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/

WHO. (2021). *Cholera Fact Sheet*. Retrieved from World Health Organization Web site: https://www.who.int/news-room/fact-sheets/detail/cholera

WHO. (2023, March 31). *Reaching vulnerable populations in Kenya's cholera outbreak*. Retrieved from WHO Africa: https://www.afro.who.int/photo-story/reaching-vulnerable-populations-kenyas-cholera-outbreak

Appendix 1: Gantt Chart



Appendix 2: Marking Guide

Strathmore University

School of Computing and Engineering Sciences

Information Systems Project Documentation Assessment Guide

Student Number(s)	134775
Working Title:	AI DRIVEN WEB-BASED APPLICATION FOR CHOLERA MANAGEMENT IN KENYA

Evaluation Points	Weight	Score	Notes
Title	1		
Abstract *Updated to include chapter 1-6*	3		
Chapter 1-3 **Checking previous proposal chapters for the correctness of title and problem statement, project scope as implemented and change of tenses*			
Problem Statement	1		
Justification	1		
Scope	1		
Limitation	1		
Literature Review	2		
Methodology	2		
Chapter 4 Correct functional requirements Correct non-functional requirements System Architecture and accompanying literature 4 Design diagrams and accompanying literature	 3 3 2 4		
Chapter 5 Setup Description: Hardware, software, support libraries, frameworks, versions, and compatibility Description of how the solution works to meet problem and business needs Description of the test environment, data, test case	 6 3 6		

<table>
<tr><td><table><tr><td>Functional Requirement</td><td>Test Data</td><td>Expected Result</td><td>Actual Result</td><td>Pass/Fail</td><td>Evidence</td></tr></table>*Check 3 core functional requirements and evidence of test available as appendix</td><td></td><td></td><td></td></tr>
<tr><td>Chapter 6
Valid Conclusion
Sound Recommendation</td><td>2
2</td><td></td><td></td></tr>
<tr><td>Presentation
Document Structure as per template provided and grammar
Citation and References
Document Numbering and Table of Contents/figures
Existence of required appendices</td><td>
2
2
2
1</td><td></td><td></td></tr>
<tr><td>Total Marks</td><td>50</td><td></td><td></td></tr>
</table>

Comments

____

____

____

____

Examiner Name: _________________________ Signature: __________________

Appendix 3: Plagiarism Report

Ouriginal
by Turnitin

Document Information

Analyzed document	134775 - IS II Project Documentation.docx (D178995456)
Submitted	2023-11-16 16:15:00
Submitted by	
Submitter email	hakeem.alavi@strathmore.edu
Similarity	3%
Analysis address	ttunduny.strath@analysis.urkund.com

Sources included in the report

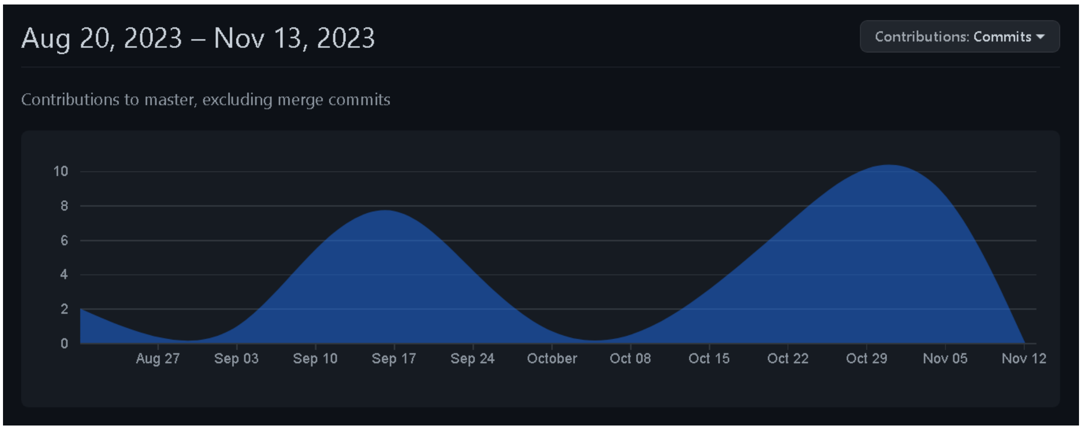
SA	**Strathmore University / W.Muchiri-2-122791 121518.docx** Document W.Muchiri-2-122791 121518.docx (D110528732) Submitted by: Emmanuel.mutiso@strathmore.edu Receiver: fsiva.strath@analysis.urkund.com	1
SA	**Strathmore University / Final Proposal write up-122285.docx** Document Final Proposal write up-122285.docx (D141811294) Submitted by: shirlyne.thiongo@strathmore.edu Receiver: fsiva.strath@analysis.urkund.com	3
SA	**Strathmore University / W.Muchiri-2-122791 121518 System Documentation.docx** Document W.Muchiri-2-122791 121518 System Documentation.docx (D114116836) Submitted by: Emmanuel.mutiso@strathmore.edu Receiver: fsiva.strath@analysis.urkund.com	3
SA	**19705F0001-p2.docx** Document 19705F0001-p2.docx (D110188733)	1
SA	**48841860.pdf** Document 48841860.pdf (D132299687)	1
SA	**Strathmore University / SHABAYA-137653-92658.docx** Document SHABAYA-137653-92658.docx (D172193209) Submitted by: Mbiyu.Alvin@strathmore.edu Receiver: fsiva.strath@analysis.urkund.com	1

Entire Document

AI DRIVEN
WEB-BASED APPLICATION FOR CHOLERA MANAGEMENT IN KENYA
ALAVI, HAKEEM FAROUK 134775

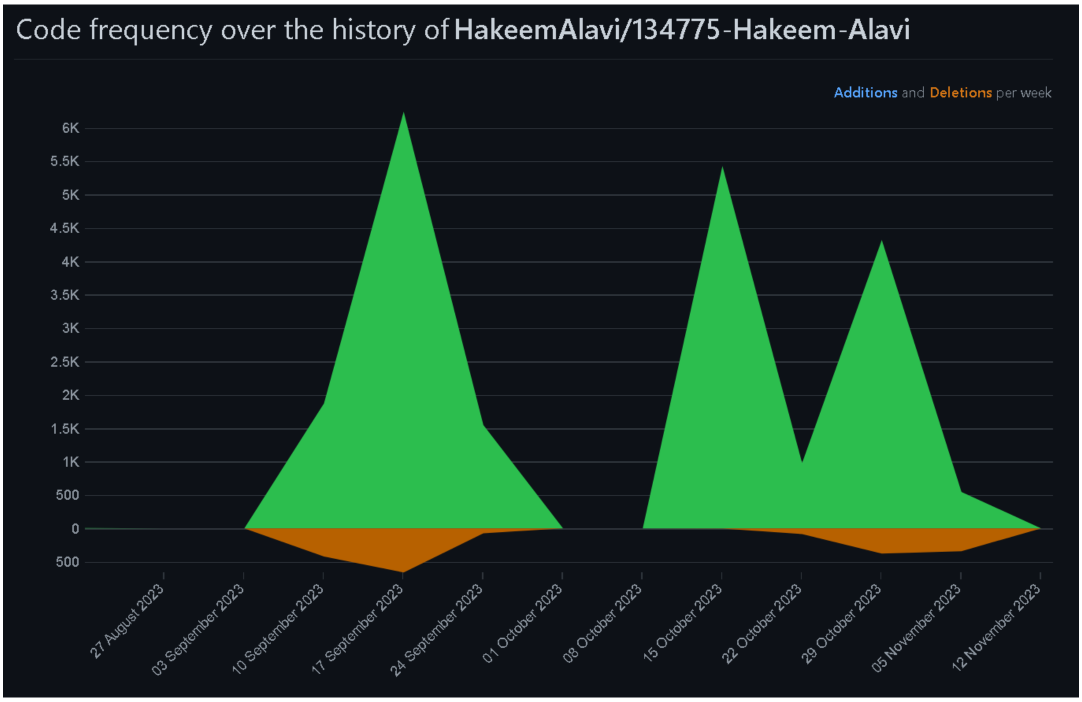
Appendix 4: GitHub Analytics and Link

Contributions Graph



The above graph visualizes the total number of contributions made in the project's repository.

Code Frequency Graph



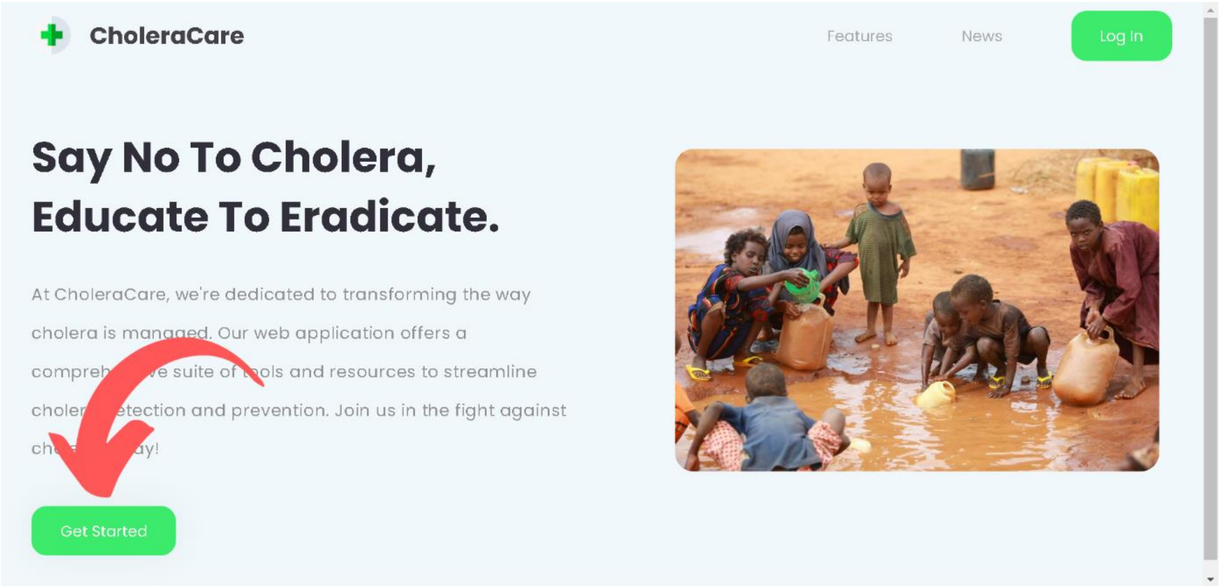
The above graph visualizes the number of additions (green in colour) and deletions (orange in colour) made as the respective contributions were made.

Repository Link

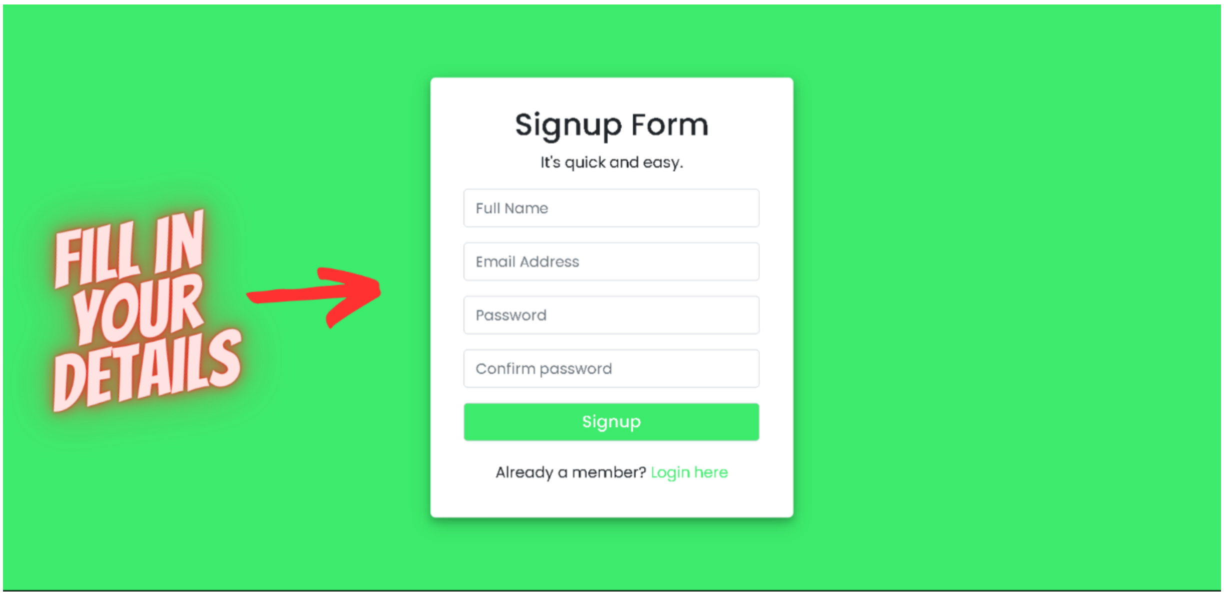
https://github.com/HakeemAlavi/134775-Hakeem-Alavi

Appendix 5: User Manual

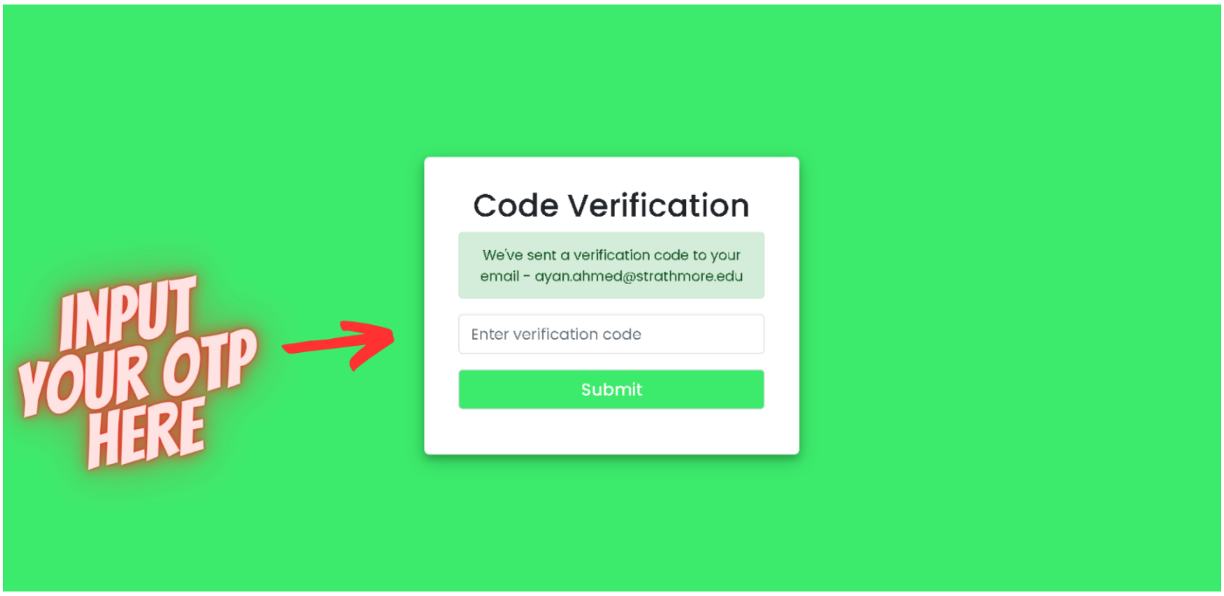
Working under the assumption that the user has already followed the developer manual, located in Appendix 6, to set up their local environment accordingly, the user can access the website link and enter it into their respective browser to access the landing page. After perusing through the landing page's content, the user can press the conveniently located call to action button shown below to sign up to the system.



The user will then be redirected to the sign-up page, shown below, where he/she should key in his/her respective details required.

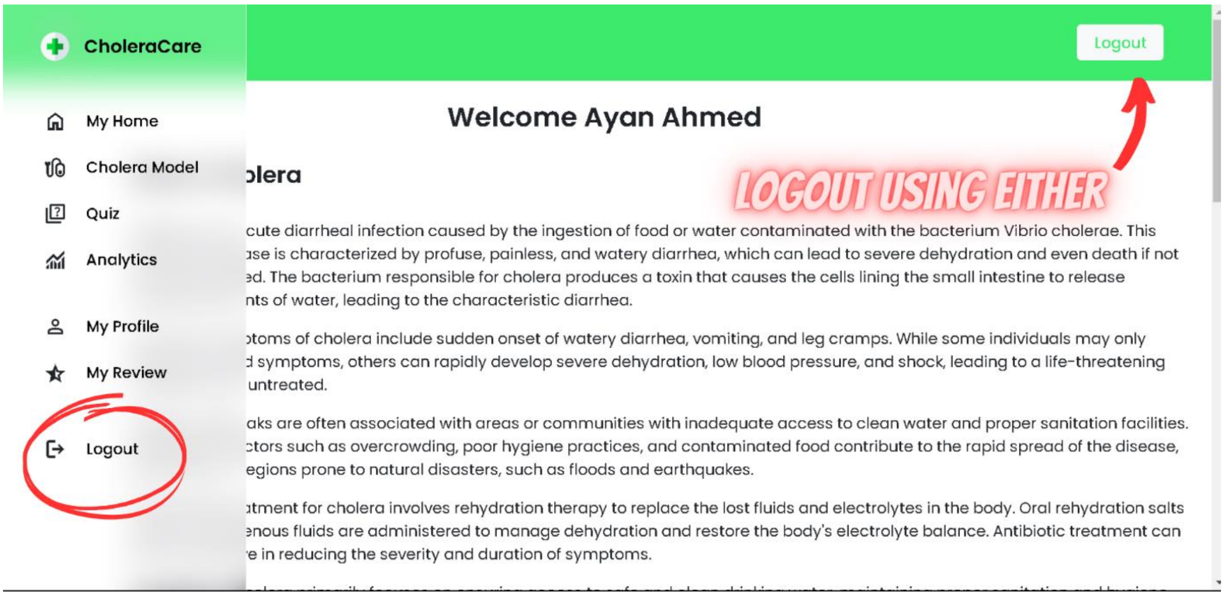


After submitting the required sign-up information, the user will be redirected to the one-time password (OTP) page where he/she will be required to input the OTP that was sent to the respective email address previously provided.



After inputting the right one-time password, the user will be redirected to his/her home page, as shown below. The side navigation panel can then be accessed by simply hovering over it causing it to expand and therefore display the relevant names of the icons thus enabling the user to browse and access the relevant features offered by the web application. The side navigation bar is sticky and therefore is always present and accessible from all pages in the user module of the web application.

After the user has been satisfied with their respective usage of the web application, he/she can logout using the top navigation bar or the side navigation panel since they both contain a logout button. Upon clicking, the user will be redirected back to the landing page.



Appendix 6: Developer Manual

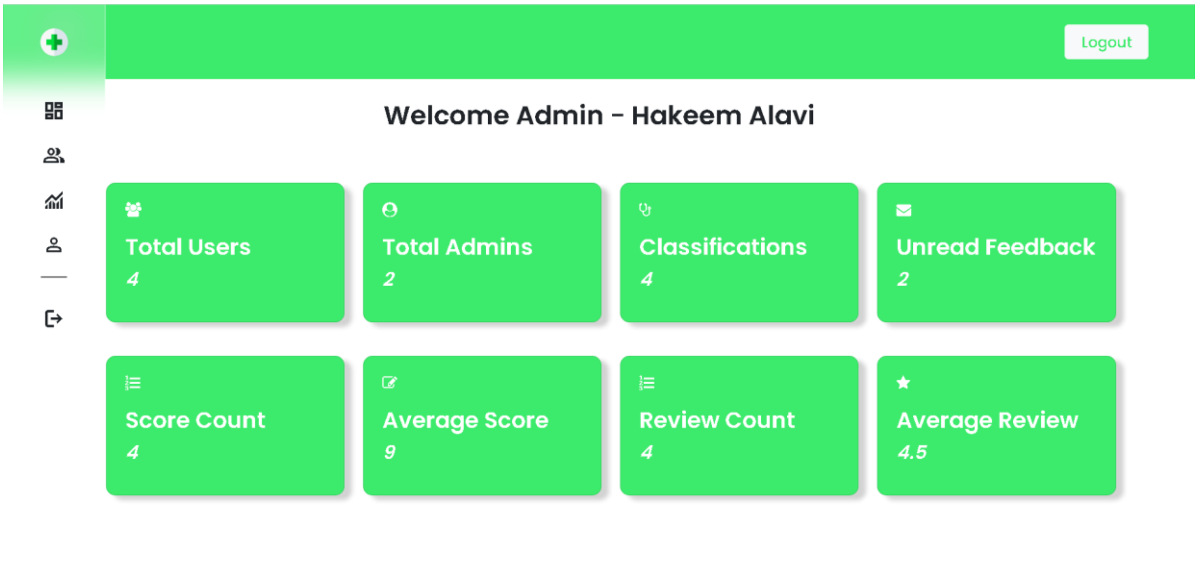
In order to run this project locally on a computer, the following are the prerequisites that should be present in the respective device (under the assumption that all system requirements are met such as a compatible OS and the presence of sufficient memory and processing power):

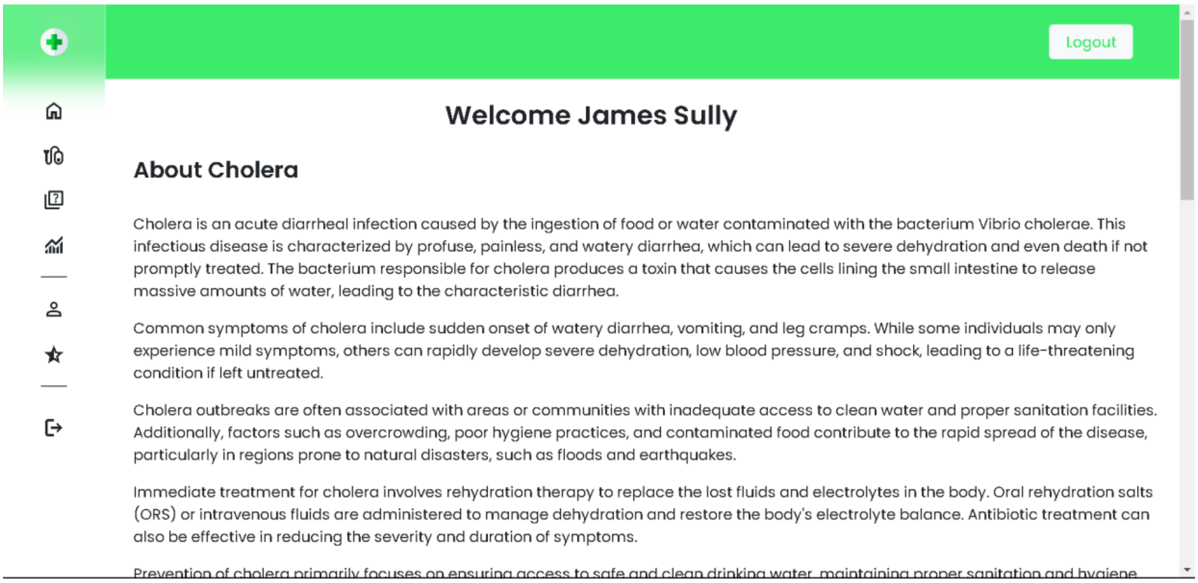
- XAMPP should be downloaded, installed, and configured appropriately
- The project's zipped folder should be downloaded from the respective GitHub repository then its contents extracted in the 'htdocs' folder located in the 'xampp' folder previously installed
- The Apache and MySQL processes within the XAMPP control panel should be started and running
- Then the PhpMyAdmin interface can be accessed where the user should have created a database called 'cholera' and then pasted the SQL code, that is located in the project folder in a file named 'cholera.sql', in the appropriate SQL section and have it run; thus, the database should have been installed locally

After the prerequisites have been met, the user can then proceed to their web browser and navigate to the local URL where the respective project application is hosted thus redirecting the user to the web application's landing page. Proceeding, the user can now fully interact with the system as intended.

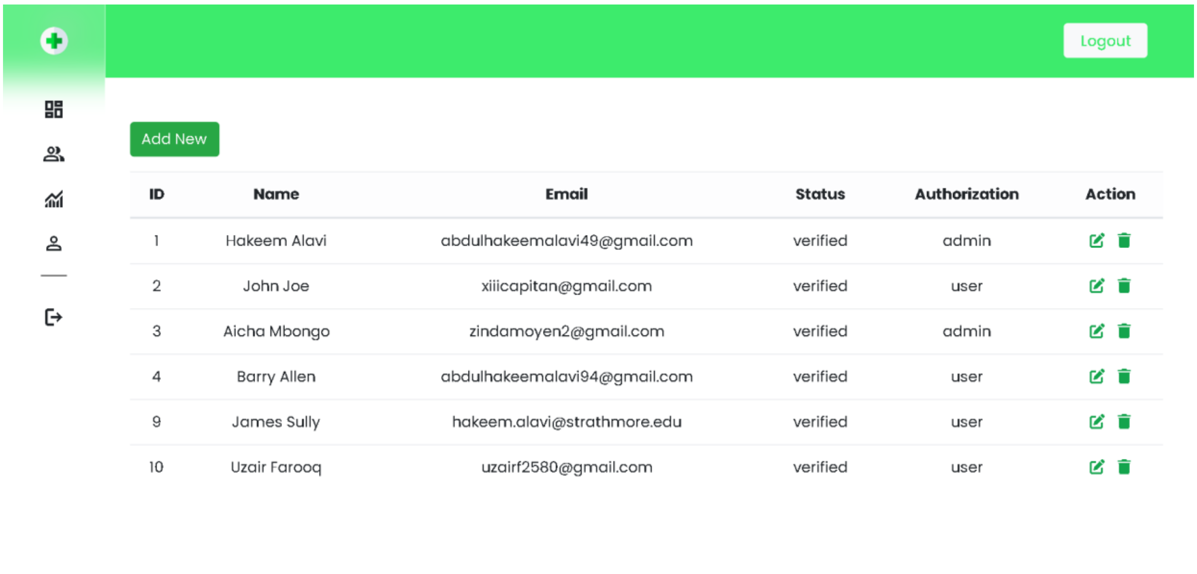
Appendix 7: Testing Evidence

i. BB01

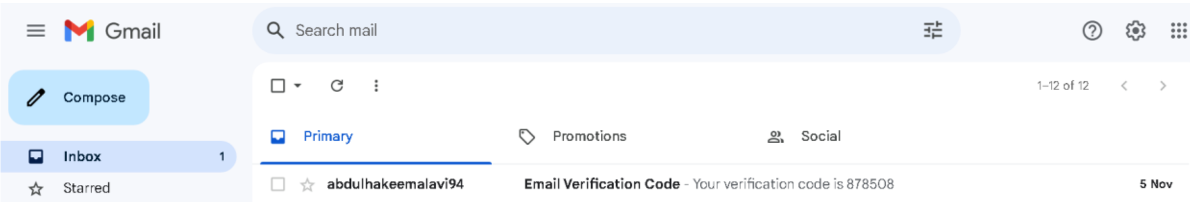




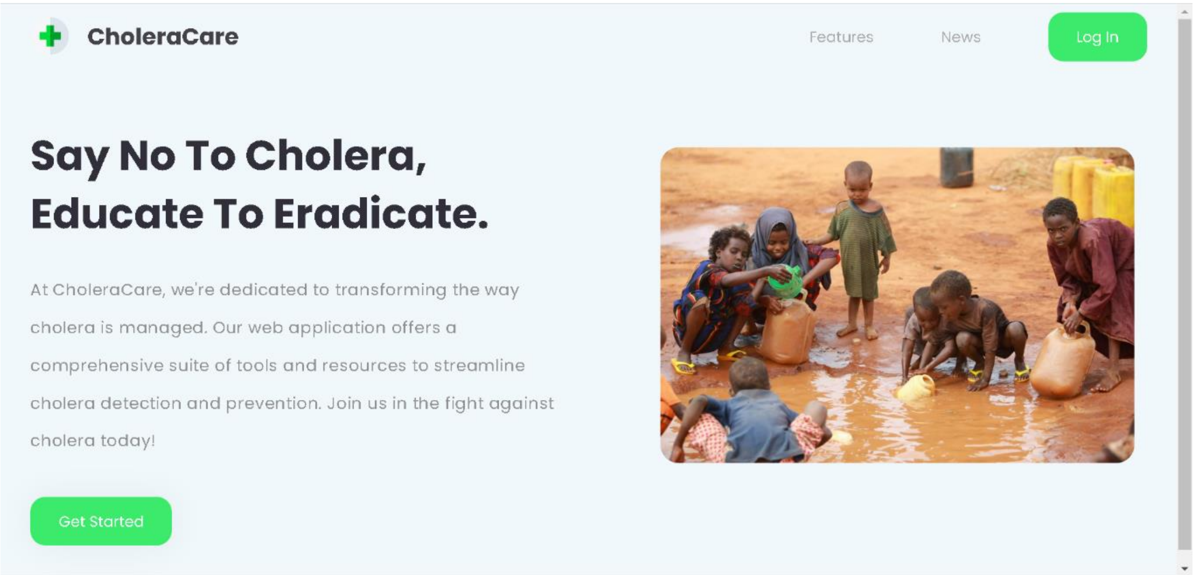
ii. BB02



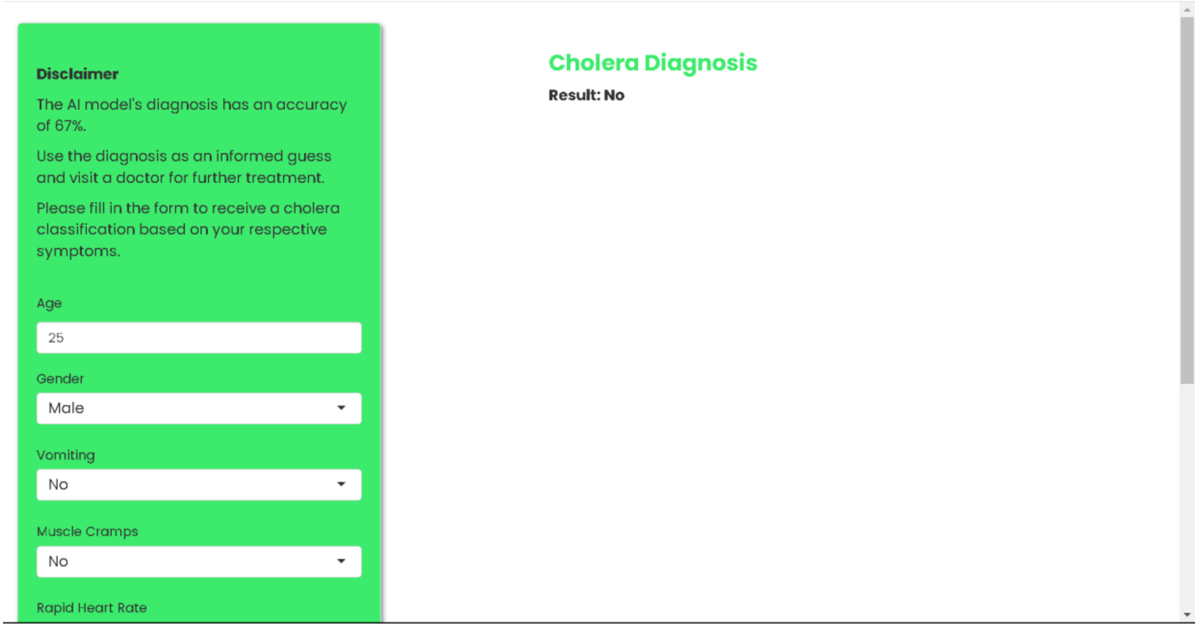
iii. BB03



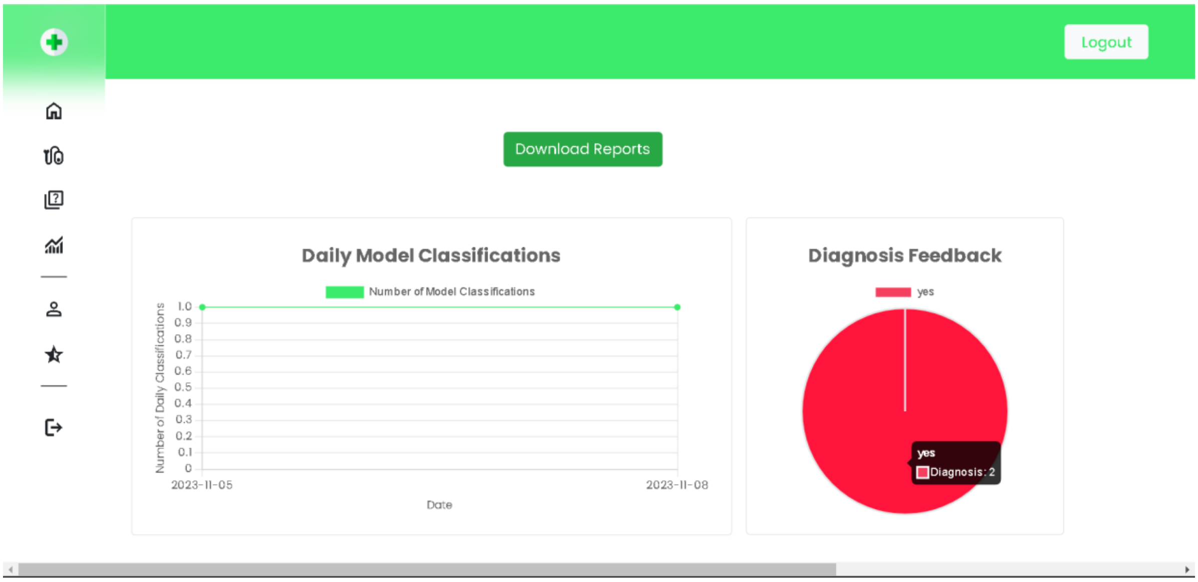
iv. BB04



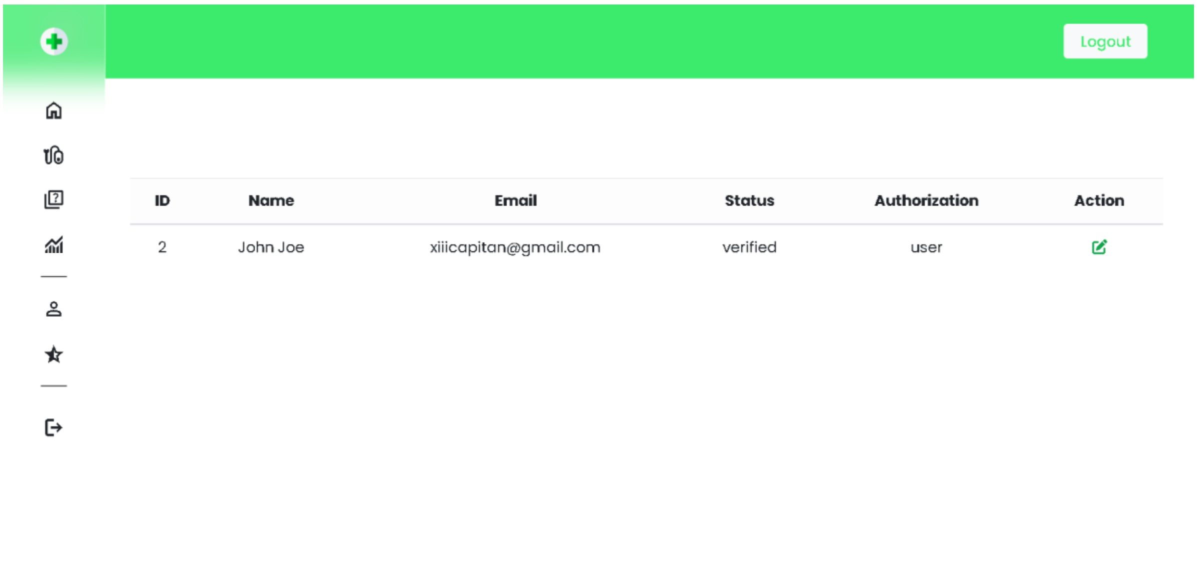
v. BB05



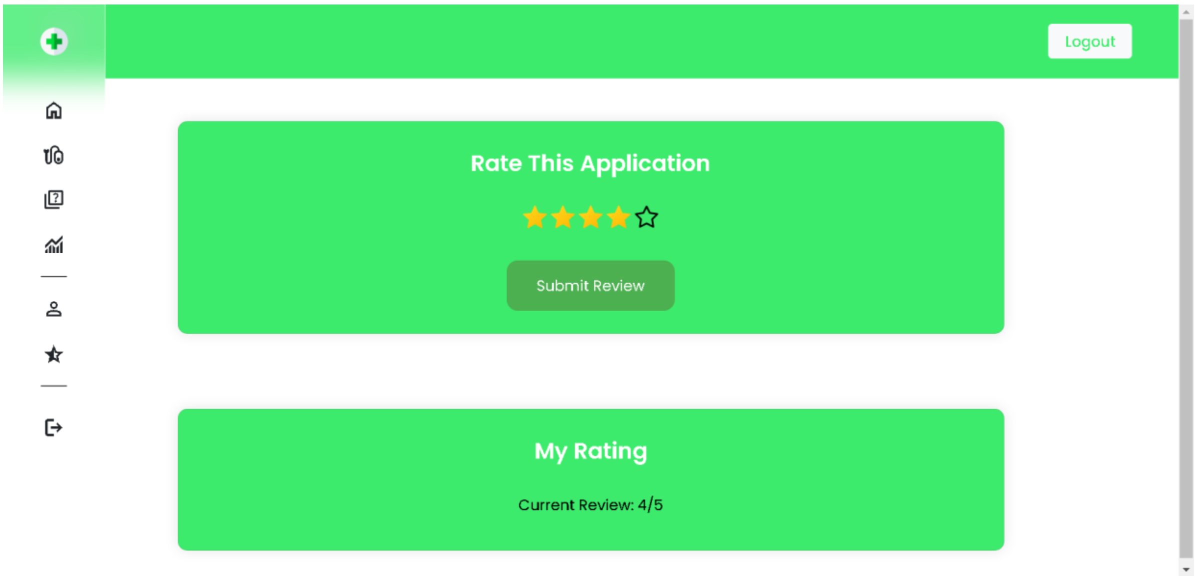
vi. BB06



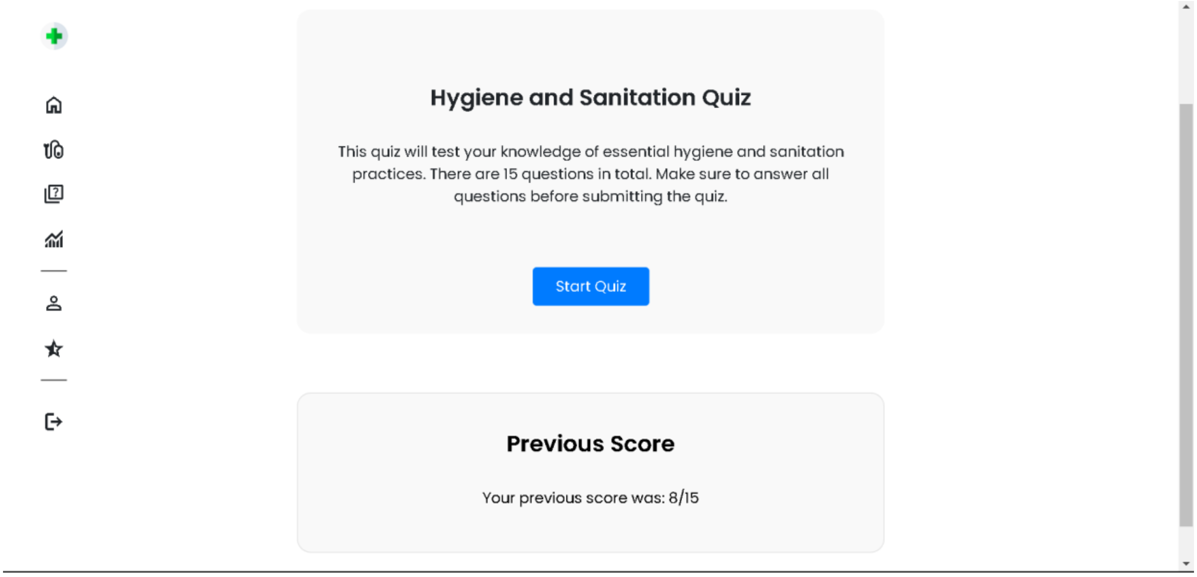
vii. BB07



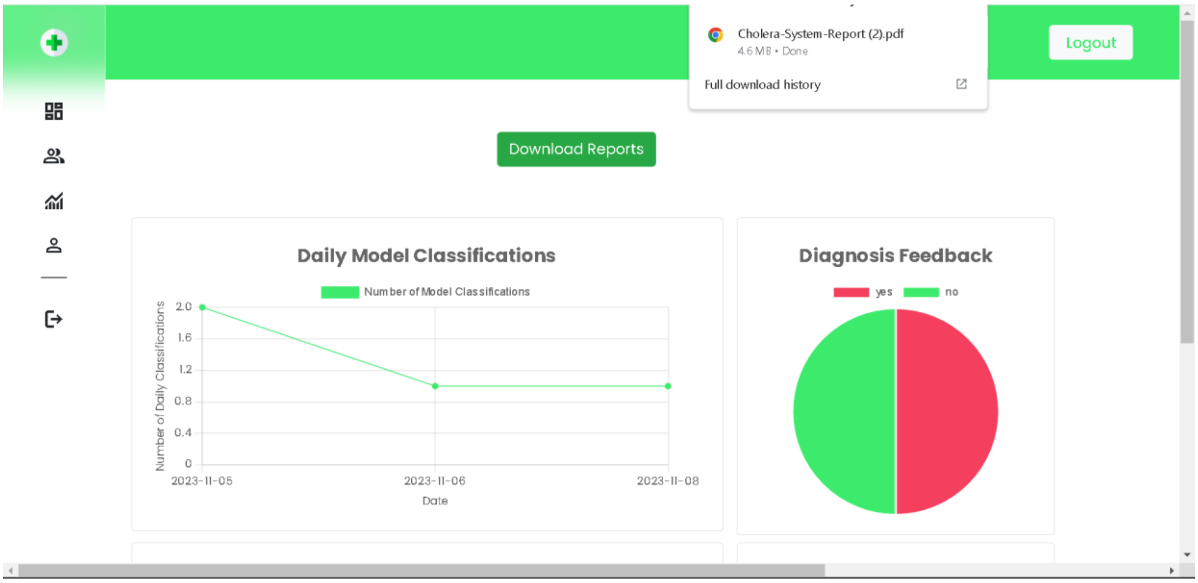
viii. BB08

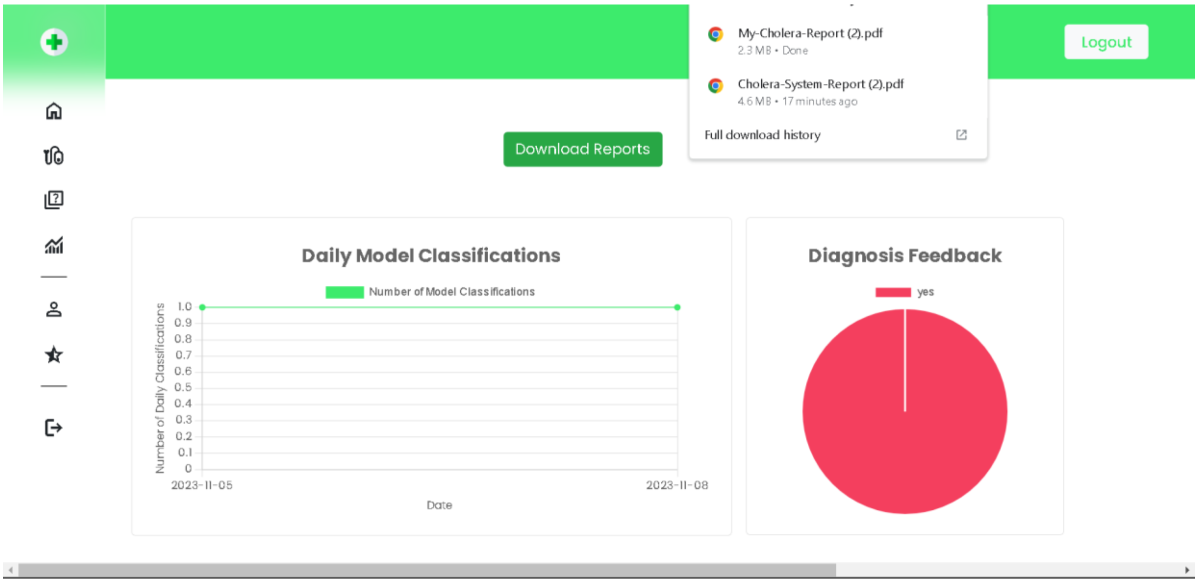


ix. BB09



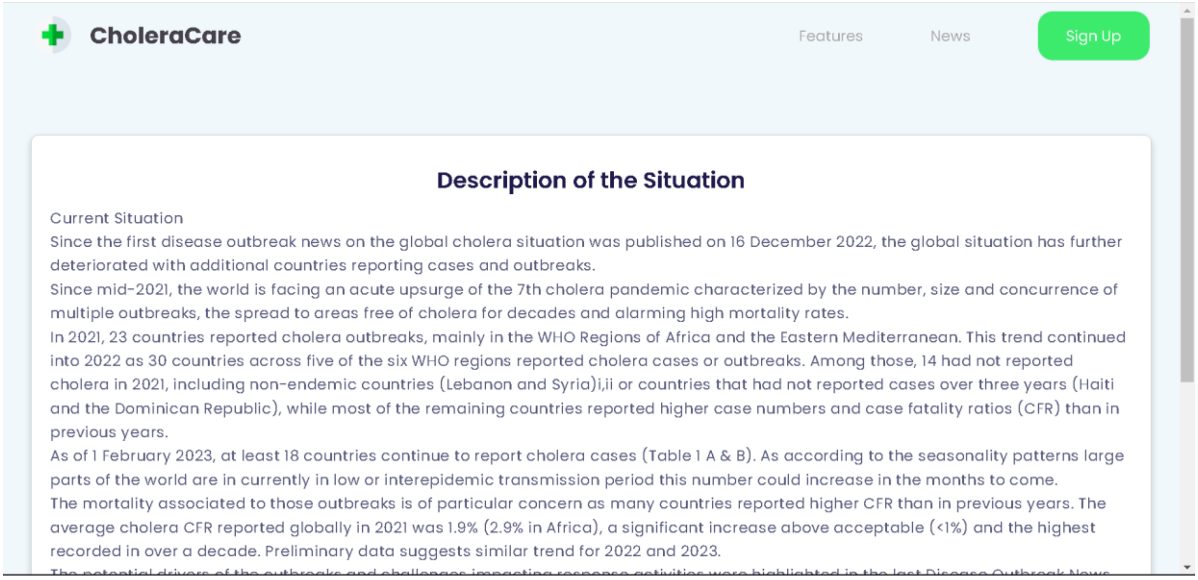
x. BB10





My-Cholera-Report (2).pdf
2.3 MB • Done
Cholera-System-Report (2).pdf
4.6 MB • 17 minutes ago
Full download history
Logout
Download Reports
Daily Model Classifications
Number of Model Classifications
Number of Daily Classifications
2023-11-05
2023-11-08
Date
Diagnosis Feedback
yes

xi. BB11



CholeraCare
Features
News
Sign Up
Description of the Situation
Current Situation
Since the first disease outbreak news on the global cholera situation was published on 16 December 2022, the global situation has further deteriorated with additional countries reporting cases and outbreaks.
Since mid-2021, the world is facing an acute upsurge of the 7th cholera pandemic characterized by the number, size and concurrence of multiple outbreaks, the spread to areas free of cholera for decades and alarming high mortality rates.
In 2021, 23 countries reported cholera outbreaks, mainly in the WHO Regions of Africa and the Eastern Mediterranean. This trend continued into 2022 as 30 countries across five of the six WHO regions reported cholera cases or outbreaks. Among those, 14 had not reported cholera in 2021, including non-endemic countries (Lebanon and Syria)i,ii or countries that had not reported cases over three years (Haiti and the Dominican Republic), while most of the remaining countries reported higher case numbers and case fatality ratios (CFR) than in previous years.
As of 1 February 2023, at least 18 countries continue to report cholera cases (Table 1 A & B). As according to the seasonality patterns large parts of the world are in currently in low or interepidemic transmission period this number could increase in the months to come.
The mortality associated to those outbreaks is of particular concern as many countries reported higher CFR than in previous years. The average cholera CFR reported globally in 2021 was 1.9% (2.9% in Africa), a significant increase above acceptable (<1%) and the highest recorded in over a decade. Preliminary data suggests similar trend for 2022 and 2023.