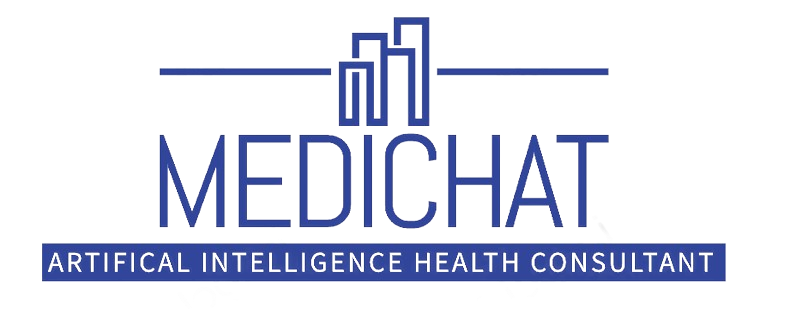
****

**Software Engineering Department**

**Braude College of Engineering**

**Capstone Project Phase B – 24-1-D16**

**MediChat - Intelligent Healthcare Consultation**

**Students:**

**David Asulin**

**Gal Danenberg**

**Supervisor:**

**Alexander Keselman**

TABLE OF CONTENTS

[**Abstract2**](#Abstract)

[**Introduction2**](#Introduction)

[**Background and Related Work2-10**](#BackgroundRelated)

[Related Work 3](#Related)

[Artificial Intelligence 4](#ArtificialIntelligence)

[Artificial Intelligence in Healthcare 4-6](#ArtificialIntelligenceInHealthcare)

[Artificial Intelligence in the Development Department](#ArtificialIntelligenceInHealthDe) 6-7

[API - Application Programming Interface 7](#API)

[Next.JS - Mongo Architecture 7](#Nextjs)

[WebSocket 8](#websocket)

[Source Control - GitHub 8](#source_control)

[Vercel Deployment 9](#vercel)

[**Expected Achievements 10-11**](#lExpectedAchievements)

[Goals10](#Goals)

[Success Criteria 10](#SuccessCriteria)

[Unique Features 11](#UniqueFeatures)

[**Engineering Process 11-22**](#EngineeringProcess)

[**Process 11-14**](#Process)

[The Problem 11](#TheProblem)

[Requirements 11](#Requirements)

[Research 12](#Research)

[Methodology 14](#Methodology)

[Maintenance Plan 14](#MaintenancePlan)

[Challenges 14](#Challenges)

[**Product 15-24**](#Product)

[Solution 15](#Solution)

[Technology Stack 15](#Technologies)

[Architecture Diagram 16](#ArchitectureDiagram)

[Use Case Diagram 17](#UseCaseDiagram)

[Activity Diagram 18](#ActivityDiagram)

[Package Diagram 19](#PackageDiagram)

[Deployment 20](#Deployment)

[User Documentation 20](#UserDocumentation)-24

[**Verification and Evaluation 25-29**](#VerificationAndEvaluation)

[Testing Plan 25](#VerificationAndEvaluation)

[Evaluation Plan 25](#Evaluation)

[Maintenance Guide 26-29](#MaintenanceGuide)

**References 30-31**

**Abstract**

The need for convenience and effectual online healthcare consultations has led to an exponential growth in the development of creative solutions using artificial intelligence. As a response, we propose the creation of a website that is easy to use so as to cater for this requirement. An AI-based doctor interface that gives correct diagnosis and relevant health information is fast being its key feature. Our platform makes use of artificial intelligence (AI) technologies such as machine learning (ML) and natural language processing (NLP) aimed at making medical advice delivery more efficient, accurate and accessible. In order to achieve full-stack web development, we have used MongoDB, Next.js, React, and Node.js commonly referred to as MERN stack within our project. For interactive conversations, ChatGPT’s API has been integrated dynamically. We want a system which serves different kinds of users while still upholding privacy as well as transparency in data usage. Some expected outcomes are good client ratings, right diagnoses given out coupled with wider availability of healthcare guidance thus contributing towards advancements made in online health consultations

1. **Introduction**

The problem at hand is about the need for accessible and efficient online healthcare consultations. Traditional medical consultations involve a lot of processes which are complex and there is a rising demand for easy-to-use platforms that allow people to conveniently seek medical advice.

Moreover, many individuals find it hard to access diagnostic services because of physical or mental conditions thereby making it difficult for them to reach healthcare providers. These barriers towards getting timely doctor’s appointments are further worsened by long queues. Although there are numerous options today that can be used to solve this problem, they often have limited effectiveness due to several reasons [13].

Many of these alternatives require payment via subscriptions which might discourage potential users who only need them occasionally. Additionally, some solutions are too advanced thus excluding a wider range of audience from using them. In some instances, there is also no support for multiple languages which is a big disadvantage. Hence our aim is to make a website that has free online service in many different languages [15].

It should be operated by an AI doctor interface which must be made simple for anyone using it even if they are not well-versed with the internet. Many people will benefit from this, including HMOs (Health Maintenance Organization) [9]. who can reduce doctors’ workload thus relieving them while increasing their clinic’s efficiency at the same time.

Moreover, there might be more customers attracted hence making the platform useful than ever. Also, ordinary people can stay home and avoid unnecessary trips to hospitals so long as they have access to this kind of platform. This is especially important because sometimes persons with disabilities may find it hard leaving their houses due to physical limitations or mental health challenges which may arise along the way.

**2. Background and Related Work**

**2.1 Related Work**

In recent years, the landscape of online healthcare consultations has witnessed a surge in innovative solutions driven by artificial intelligence (AI) and chatbot technologies. The prevalence of traditional medical consultation models, characterized by complexity and physical presence, has prompted the exploration of alternative approaches to meet the evolving needs of healthcare consumers [14].

**Symptom Checking Solutions:**

One notable solution in this domain is **Ada** [10], which utilizes AI to assess symptoms, provide personalized support, and connect users with relevant mental health resources. Ada's strengths lie in its evidence-based content and options for anonymity. However, it relies on self-reported data and lacks direct clinical intervention, posing certain limitations.

Another solution, **Your.MD** [11], combines a chatbot interface with a physician network to provide symptom checking, virtual consultations, and prescription refills. While it grants access to qualified professionals, potential cost barriers and limited availability of healthcare providers pose challenges.

**Relevance and Importance of Proposed Solution:**

The growing demand for user-friendly platforms and the increasing reliance on AI technologies in healthcare underscore the relevance and importance of our proposed solution. By examining past approaches and existing solutions, we can identify key issues and challenges in the domain of online healthcare consultations. These may include limitations in accessibility, concerns regarding the accuracy and reliability of AI-driven diagnoses, and ethical considerations surrounding the use of patient data [7].

**Proposed Solution:**

Building upon the insights gained from past research and existing solutions, our proposed solution aims to offer a user-friendly platform for online healthcare consultations, supported by an AI-driven interface. By leveraging advancements in AI technology, such as natural language processing (NLP) and machine learning, we seek to enhance accessibility, accuracy, and efficiency in medical advice delivery. Furthermore, by incorporating multilingual support and prioritizing user experience, we aim to overcome the limitations of existing solutions and provide a comprehensive healthcare solution that caters to diverse user needs.

### **2.2 Artificial Intelligence**

**2.2.1 ChatGPT**

Launched in November 2022, ChatGPT represents a breakthrough in AI technology, developed by OpenAI as a freely accessible large language model chatbot. Harnessing the power of GPT-3.5, it serves as a versatile tool capable of engaging in conversations, answering queries, and executing tasks upon request. With its adaptability, users can tailor ChatGPT to suit their specific needs, offering a glimpse into the future of human-computer interaction. Furthermore, the integration of GPT-4 technology promises to enhance the quality and functionality of ChatGPT, paving the way for even more personalized and insightful interactions. Notably, ChatGPT not only provides answers but also offers "justifications," often containing novel perspectives that can inspire further research endeavors [1].

**2.2.2 Artificial Intelligence Conversational Bots**

The evolution of chatbots has led to a diverse range of architectures and functionalities, expanding their utility across various domains. These AI-driven agents excel in understanding user input and delivering meaningful responses by leveraging pre-existing knowledge. Particularly in healthcare, AI chatbots play a vital role in augmenting patient care by facilitating symptom checking, appointment scheduling, medication reminders, and even virtual consultations. As the capabilities and architectures of chatbots continue to evolve, their applications across different sectors continue to expand, driven by their ability to comprehend and respond to user queries based on compiled knowledge [3].

**2.3** **Artificial Intelligence in Healthcare**

**2.3.1 Advancements in AI-Powered Healthcare Solutions**

Advancements in AI-powered healthcare solutions have gained significant momentum, with ChatGPT emerging as a notable example of natural language processing (NLP) technology in the medical field. While ChatGPT has shown potential in various healthcare applications, including medical education, patient communication, and clinical decision support, it is important to approach its use with caution [1]. The systematic review reveals that ChatGPT has achieved moderate or 'passing' performance in various medical tests, but it is not yet reliable for actual clinical deployment [1]. Specialized NLP models trained on biomedical datasets are currently more suitable for critical clinical applications. As AI continues to evolve, researchers are exploring ways to leverage large language models (LLMs) like ChatGPT to enhance healthcare delivery, streamline workflows, and improve patient outcomes. However, ethical considerations, accuracy concerns, and the need for domain-specific training remain crucial challenges to address before widespread implementation in healthcare settings [1].

**2.3.2 Applications in Healthcare**

AI has found widespread applications in healthcare, promising to revolutionize patient care and optimize medical processes. AI-powered chatbots and virtual assistants serve as valuable resources, providing round-the-clock support, facilitating diagnosis, and enhancing treatment planning. [12]

**2.3.3 Ethics problem in Artificial Intelligence**

The integration of artificial intelligence (AI) in healthcare raises significant ethical concerns that must be carefully addressed. One of the primary issues is the potential for algorithmic bias, where AI systems may perpetuate or amplify existing societal biases based on factors such as race, gender, or socioeconomic status [7]. This bias can lead to unfair and discriminatory outcomes in healthcare, such as unequal access to treatment or inaccurate diagnoses for certain patient populations [7]. Additionally, the use of AI in sensitive medical decisions raises questions about accountability and liability, particularly when AI systems make errors or cause harm [7]. There are also concerns about data privacy and the appropriate use of patient data in AI development and deployment [7]. As AI continues to advance in healthcare, it is crucial that researchers, healthcare providers, and policymakers work together to establish robust ethical frameworks and guidelines to ensure the responsible and equitable use of this technology [7]. This includes promoting transparency, explainability, and oversight in AI systems, as well as involving diverse stakeholders in the decision-making process [7]. By proactively addressing these ethical challenges, we can harness the potential of AI to improve healthcare while safeguarding patient rights and promoting social justice.

**2.3.4 Case Studies**

**2.3.4.1 TidalHealth Peninsula Regional**

TidalHealth [16] worked with IBM to implement IBM Micromedex with Watson AI, a cloud-based clinical decision support (CDS) system. The system combined several capabilities for TidalHealth: it aggregated clinical information from various sources, including drugs, diagnose, and therapeutic procedures; used natural language processing (NLP) and machine learning (ML) to understand user queries and provide relevant, evidence-based recommendations; and integrated with TidalHealth's electronic health record (EHR) system.

Results

* Clinicians reported saving up to 20 minutes per encounter
* Increased adherence to best practices
* Increased consistency in care

**2.3.4.2 Portal Telemedicina**

Portal Telemedicina [17] worked with Google Cloud on the health provider’s data aggregation, storage and analysis. Cloud SDK routes data through a gateway to cloud storage. AI classifies medical findings and recommends treatment urgency, such as analyzing chest X-rays for pneumonia.

Results

* 20% reduction in hospital admissions
* 5% reduction in health care costs
* Improved patient outcomes

**2.3.4.3 Amsterdam UMC**

“Our opportunity is to use AI to help us with our ever-growing data volumes,” says Dr. Geert Kazemier, professor of surgery and director of surgical oncology, Amsterdam UMC [18].

The team at Amerstam worked with SAS and its AI-based analytics platform: they integrated data from diverse sources for a unified environment; provided advanced generative AI analytics tools for research and predictive modeling; and clinicians and researchers have real-time access to critical data.

Results

* Improved patient care through personalized treatments based on data insights
* Innovative research capabilities through advanced analytics

**2.4 Artificial Intelligence in Development Department**

**2.4.1 Integration of ChatGPT and REACT**

The integration of the ChatGPT API with React presents an exciting opportunity to create dynamic and interactive user experiences. React is a popular JavaScript library for building user interfaces, known for its efficient rendering and ability to manage complex state. By leveraging React's component-based architecture and robust state management capabilities, developers can seamlessly incorporate ChatGPT's natural language processing functionalities into web applications [2]. Through the API, React components can interact with ChatGPT in real-time, facilitating the creation of chatbots, virtual assistants, and advanced conversational interfaces. This integration not only enhances user engagement but also demonstrates the potential for synergistic collaboration between AI technologies and web development frameworks like React [2].

**2.4.2 Innovations in AI-Web Integration**

Integrating AI technologies like ChatGPT with web development frameworks like React opens up new avenues for innovation. By combining AI-driven NLP with dynamic user interfaces, developers can create immersive applications that enhance user engagement and accessibility. This integration also offers opportunities to optimize performance and improve user experience by leveraging React's state management capabilities.

**2.4.3 User-Friendly UI in AI Applications**

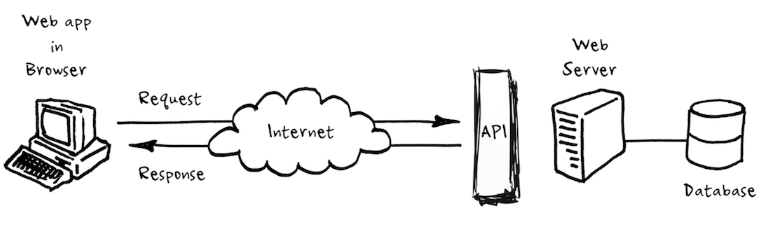
User-friendly UI design is crucial in the development of AI applications to ensure seamless interaction between users and the AI system. In AI applications, the UI serves as the interface through which users input queries, receive responses, and interact with AI-driven functionalities. A user-friendly UI design focuses on simplifying complex AI functionalities, making them accessible and intuitive for users of varying technical expertise.

The UI should provide clear and concise instructions on how to interact with the AI system, guiding users through the process effectively. Visual cues, such as progress indicators or loading animations, can help users understand when the AI system is processing their requests.

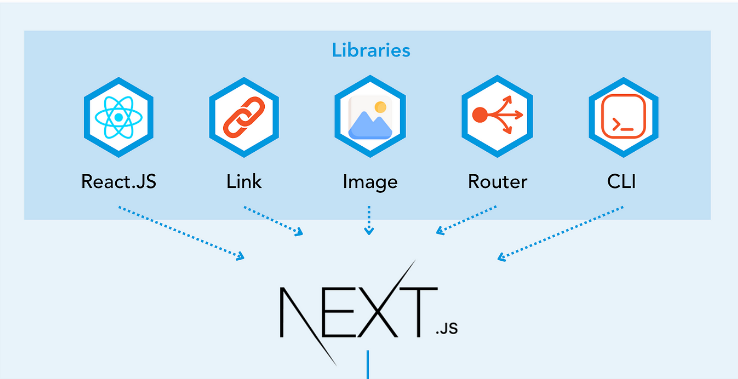
**2.5** **API - Application Programming Interface**

An API (Application Programming Interface), is a set of rules and protocols that allows different software applications to communicate and interact with each other.

APIs are central to many modern software architectures, as they provide high-level abstractions that facilitate programming tasks, support the design of distributed and modular software applications and the reuse of code[8]. It essentially defines how different components of software systems can interact, enabling them to exchange data and perform various functions. APIs are commonly used to enable integration between different software systems, allowing them to work together seamlessly. They provide a standardized way for developers to access specific features or functionality of a software application, without needing to understand the underlying code.

**2.6** **Next.JS - Mongo** **Architecture**

Our background research led us to transition from the MERN stack to Next.js for several key reasons. Built on top of React, Next.js enhances the development process by offering server-side rendering, static site generation, and API routes out of the box, which significantly improves performance, and overall development experience. Its hybrid nature allows for the combination of static and dynamic content, fulfilling a wide range of application requirements. The framework's built-in routing system and extensive optimization features reduce the need for additional configuration. This transition aligns with our project goals of delivering high-performance, scalable, and maintainable web applications, making Next.JS the optimal choice for our evolving needs.



**2.7** **WebSocket**

WebSocket is a computer communication protocol that provides a full-duplex communication channel over a single TCP connection. It is designed to be implemented in web browsers and web servers, but it can be used by any client or server application. WebSocket was developed as a way to enable real-time communication between web applications and servers. It allows a client (such as a web browser) to initiate a connection to a server and then send and receive messages over that connection in real time, without the need to send HTTP requests and wait for responses. WebSocket uses a handshake process to establish a connection, after which it can send and receive data as a stream of messages. It supports both text and binary data, and it can be used for a wide range of applications, including chat, multiplayer games, real-time data visualization, and more.



**2.8** **Source Control - GitHub**

GitHub is a developer platform that allows developers to create, store, manage and share their code. It uses Git software, providing the distributed version control of Git plus access control, bug tracking, software feature requests, task management, continuous and continuous development and helps maintain versions and version updates for projects.



**2.9** **Vercel - Deployment**

Vercel deployment provides an easy way to host web applications, giving developers a seamless experience from code to production. This cloud platform uses its global edge network to ensure static content delivery and optimal performance[19], it specializes in deploying static serverless applications Vercel integration with popular version control systems like GitHub, GitLab, and Bitbucket enables automated deployments triggered by code push, simplifies Continuous Integration and Deployment (CI/CD) The workspace of the platform's preview deployment feature lets developers test changes in isolated environments before deployment collaboration, increasing productivity and reducing the risk of bugs [19] in live environments with a focus on developer experience and productivity, Versailles has become the go-to solution for modern web application teams several.

A black text on a white background

Description automatically generated

**3.** **Expected Achievements**

The website aims to provide a professional medical consulting service characterized by precision in analyzing health conditions and comprehensibility to a diverse audience. Efforts will be made to ensure that the website maintains a high level of user-friendliness and clarity, with support of multiple languages to enhance accessibility. Additionally, an accessibility menu will be incorporated into the website interface to further facilitate ease of navigation and usage for individuals with diverse needs and preferences.

The project will adhere to the Declaration of Helsinki's ethics guidelines, ensuring the well-being and rights of participants. Informed consent will be obtained, benefits will outweigh risks, and privacy will be safeguarded. An independent ethics committee will review the research plan, with ongoing monitoring of ethical considerations.

**3.1** **Goals:**

Our goal is to develop a cross-platform website optimized for computers, smartphones, and tablets, featuring the following key attributes:

* Support for multiple languages to cater to diverse user demographics.
* Implementation of an accessibility menu catering to users with disabilities.
* Successfully create a user-friendly website interface.
* Integrate an AI doctor interface capable of providing health information and diagnosis.
* Establish the platform as a free service for users.
* Seamless usability across mobile and tablet devices for enhanced convenience.
* An intuitive and understandable interface designed to accommodate users of all backgrounds and technical competencies.

**3.2** **Success Criteria:**

The project will count as successful if the following achieved:

* Positive feedback garnered from a diverse range of users representing various demographics and user profiles.
* The AI medical assistant consistently provides accurate diagnoses and effective solutions to users' health concerns.
* Users report ease of understanding and navigation, indicating intuitive usability of the website.
* The website experiences significant growth in user traffic and engagement, reflecting its increasing popularity and usefulness.
* Users find the platform to be a convenient alternative to traditional healthcare consultations, eliminating the need to leave their homes and endure long waiting times for reliable medical advice.
* No monetary charges for accessing healthcare advice and information on the platform, ensuring accessibility for all users regardless of financial status.

**3.3** **Unique Features:**

Our platform will be distinguished by several unique features not commonly found in comparable systems:

* The platform will offer multi-language support, accommodating a diverse global user base.
* Access to the platform will be provided at no cost to the user.
* The consultancy service will incorporate advanced image recognition capabilities.

In summary, our website will uniquely provide multilingual support, complimentary access, and sophisticated image recognition technology in the medical consultancy domain.

**4.** **Engineering Process**

This section elaborates about the challenges we faced and are expecting to encounter, as well techniques we had used during the engineering process of the product.

**4.1** **Process**

At the beginning, we learned about what API’s are available today and commonly used to develop such projects. We chose to work with ChatGPT’s API because it is simpler, free, and commonly used.

After we established the API, we learned about what web development stack we should use.

We chose to work with the MERN stack which consists of MongoDB, Express.js, React, and Node.js. It's a popular stack for building full-stack applications using JavaScript across the front-end and back-end.

We chose MERN because it is a well known recommended popular stack.

After we were done deciding what tools and API we will use to build the project, we conducted research on the subjects.

**4.1.1** **The Problem**

We first faced the growing requirement for accessible and efficient consultations related to health online when our initial challenge came up. Traditional method of consulting a doctor was often too complex and took much time hence making it challenging for a person to seek immediate medical advice, more so those who had physical or psychological hindrances. Though the alternatives were there, they were often ineffective because one could not access them easily due to subscription charges, or language differences.

**4.1.2** **Requirements:**

We conducted thorough research to define both functional (FR) and non-functional requirements (NFR) to ensure the system's effectiveness, usability, security, and scalability. Our FRs focus on the core functionalities such as account creation, login mechanisms, profile management, chat functionalities, and accessibility features. These were derived from user needs and industry best practices to provide a seamless user experience. Meanwhile, our NFRs address critical aspects like security, usability, reliability, and deployment.

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement** | | **FR/NFR** | **Type** |
| 1 | The system will allow guest users to create an account | FR |  |
| 1.1 | The registration processes should be intuitive and user-friendly to facilitate easy account creation. | NFR | **Usability** |
| 1.2 | User passwords must be securely encrypted/hashed and stored in the database to prevent unauthorized access. | NFR | **Security** |
| 2 | The system will include a login mechanism for registered users | FR |  |
| 2.1 | Provide clear error messages to guide users in case of incorrect login credentials. | NFR | **Usability** |
| 2.2 | The login processes should be intuitive and user-friendly to facilitate easy account access. | NFR | **Usability** |
| 3 | The system will allow registered users to retrieve / change lost password | FR |  |
| 3.1 | The password retrieval will be done with registered email / phone-number |  | **Usability** |
| 4 | The system will allow users to view their profile | FR |  |
| 4.1 | The user profile will include his personal medical data | NFR | **Data Integrity** |
| 5 | The system will allow users to edit their account details | FR |  |
| 6 | The system will allow users to edit their personal details | FR |  |
| 7 | The system will allow users to edit their health metrics details | FR |  |
|  | Editing and Viewing personal information will require re-entering accounts password for extra safety | NFR | **Security** |
|  | Ensure that sensitive user information, such as personal details, is protected from unauthorized access or disclosure. | NFR | **Security** |
| 8 | The system will allow users to start new chat with the AI system | FR |  |
| 8.1 | The chat will support multiple languages | NFR | **Usability** |
| 9 | The system will allow users to provide images to the AI | FR |  |
| 10 | The system will allow users to continue/view existing chats with the AI system from the history | FR |  |
| 10.1 | Viewing History will be available only for registered users | NFR | **Reliability** |
| 11 | The system will allow users to delete existing chats with the AI system from the history | FR |  |
| 12 | The system will allow users to access accessibility menu | FR |  |
| 12.1 | The accessibility menu will include changing the text size to different sizes for people who are visually impaired | NFR | **Usability** |
| 12.2 | The accessibility menu will include an option to change the screen contrast | NFR | **Usability** |
| 13 | The system will allow users to change the website language | FR |  |
| 14 | The system will allow users to change the font size | FR |  |
|  | The application will be deployed using Vercel for better scalability and availability | NFR | **Deployment** |
|  | The website will be accessible from every operation system / mobile that supports web applications | NFR | **Platform Compatibility** |

**4.1.3** **Research**

We conducted research on the following subjects throughout the project :

**Artificial Intelligence in Healthcare:**

* Artificial Intelligence in Health Department
* AI-Powered Healthcare Solutions
* AI Healthcare Case Studies
* Artificial Intelligence Conversational Bots
* Existing Solutions

**Technology Stack:**

* Chat GPT API
* MongoDB
* React.js
* Integration of ChatGPT and React.JS
* Node.js
* Next.js
* Vercel
* GitHub Version Control

**User Experience (UX) and Design:**

* User Experience (UX) Research
* User Friendly Design Patterns

**Ethical and Legal Considerations:**

* Ethical and Legal Considerations of AI In Healthcare

**Scalability and Infrastructure:**

* Vercel Deployment
* Scalability Solutions

**4.1.4** **Methodology**

We decided to work with the “Top-down methodology” which means that we first conduct a review of all the components in the project, that way we can study each component.

This approach significantly enhances organization and simplifies the monitoring of the learning process for each component.

In addition to employing the "Top-down methodology," incorporating elements of the Scrum methodology can further enhance project management practices. Scrum is an iterative and incremental agile framework that fosters collaboration, transparency, and adaptability in software development projects.

By integrating Scrum practices, such as daily stand-up meetings, sprint planning, and retrospectives, teams can establish a more dynamic and responsive workflow.

**4.1.5** **Maintenance Plan**

* **Bug Fixes, Version Control and Release Management:**

With each new feature and each new code addition, we might face conflicts, bugs and issues, in order to solve this we decided to maintain the code using github for version control and issues tracking

* **Performance Optimization**

We will conduct continuous testing and monitoring to find and optimize our project with PlayWright and Pytest, to locate regressions and monitor performance.

**4.1.6** **Challenges**

Initially, we confronted the challenge of developing an AI chat system that surpasses existing online solutions in accuracy and innovation.

Upon defining our project's objectives, we engaged in a thorough discussion regarding the selection of a development stack, carefully weighing the advantages and disadvantages of each option.

Given the emerging nature of artificial intelligence, particularly in the medical sector, we found that relevant information was limited.

Consequently, we conducted extensive research and reviewed analogous projects to enhance our understanding.

In the end, we decided to use containerization to organize our project well in order to reduce issues with dependencies.

**4.2** **Product**

**4.2.1** **Solution:**

We suggested making a website that can respond adequately to this particular issue. Such a platform will be user-friendly with a doctor interface that uses artificial intelligence (AI) for correct diagnosis and necessary health information provision. Machine learning and natural language processing through the MongoDB, Express, React, form some of the ways through which this can be achieved.

**4.2.2** **Technologies:**

We selected MongoDB because of its adaptability and scalability in dealing with different data types.

Next.js simplifies server-side application development with a strong framework.

React enables dynamic user interfaces by using a component-based structure.

Node.js enables smooth communication between the server and client through event-driven architecture and non-blocking I/O

And Vercel for deployment which allows us to monitor, scale, and deploy our web-app in the best possible way.

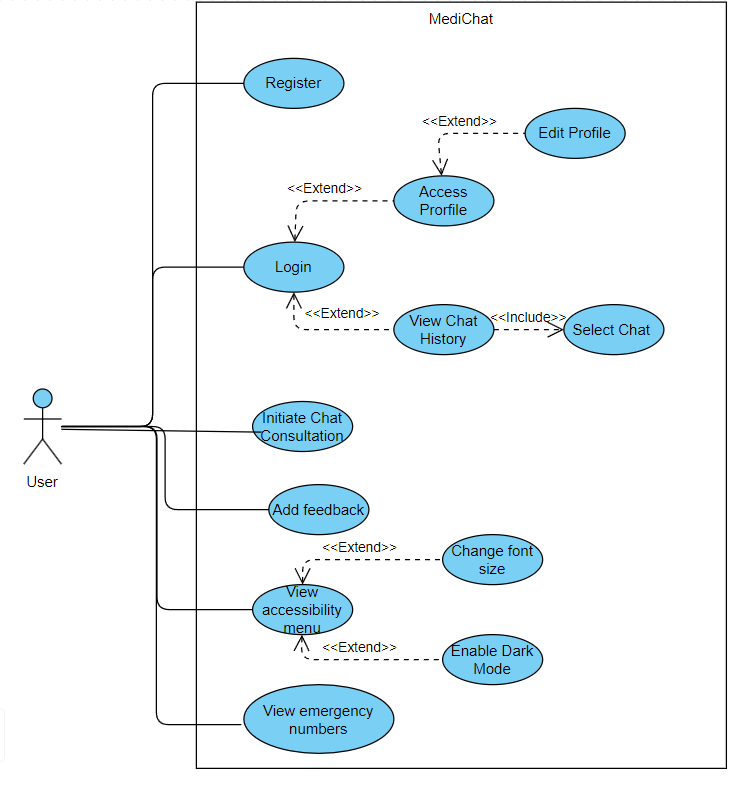
These technologies together form an efficient, scalable and flexible full-stack web development solution that perfectly matches our project objectives.

The main flow for a user interacting with our chat interface :

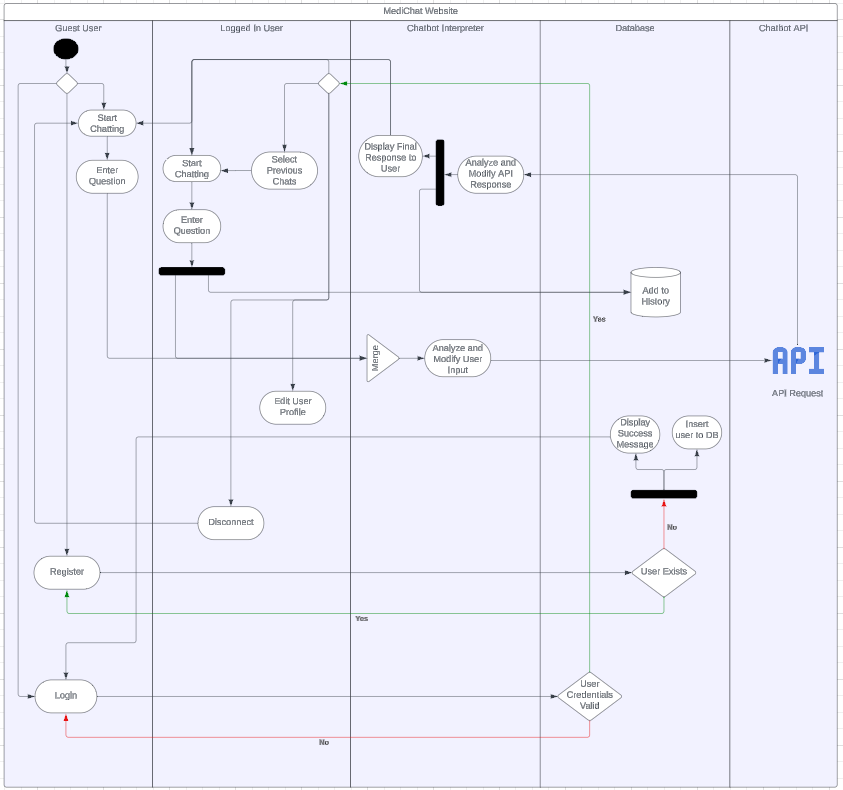
1. User enters input question to the chat
2. The user input is sent to the interpreter layer
3. The user input is being interpreted and modified in the best way for GPT/AI to return a useful and precise response
4. The interpreter layer sends the modified message to GPT/AI and retrieves the response
5. The response is being interpreted and modified and simplified for the user to understand the response
6. The response is being displayed to the user
7. Repeat 1-6.

**A diagram of a server

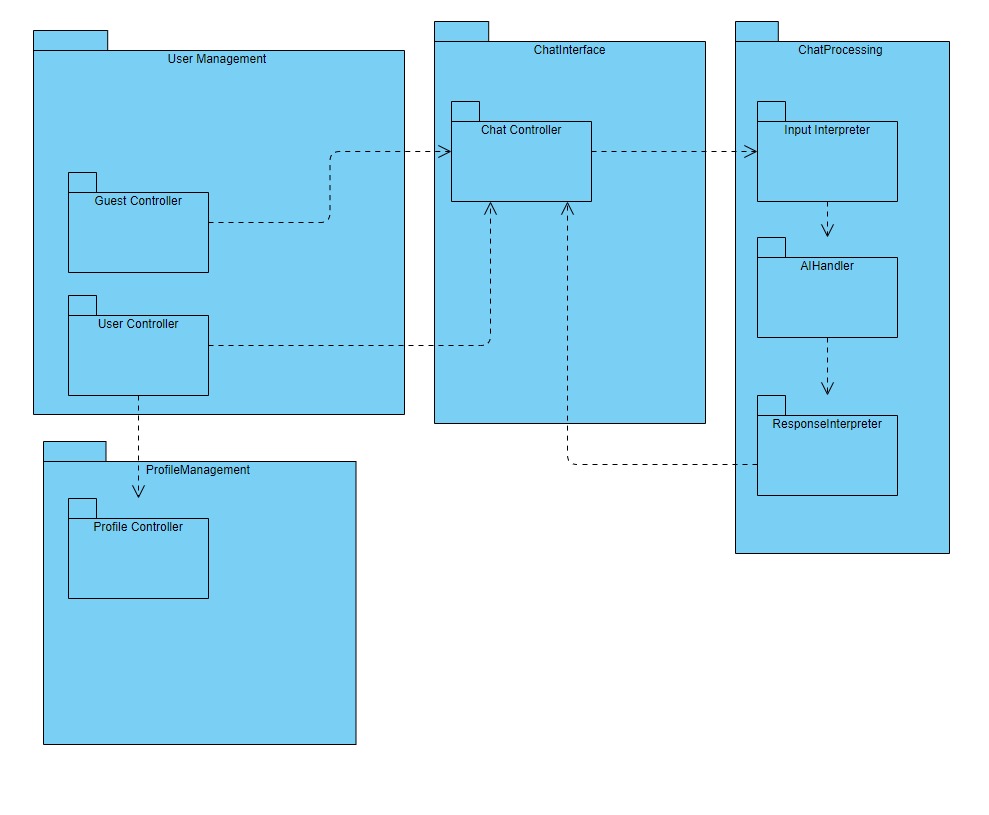
Description automatically generated4.2.3** **Architecture Diagram:**

**4.2.4** **Use Case Diagram:**

**4.2.5** **Activity Diagram:**



**4.2.6** **Package Diagram:**



**4.2.7** **Deployment:**

For developers :

Developers will be able to run the project locally by cloning our github repository with react using

* **npm run start** for frontend
* **node server.js** for backend

For users :

The website will be accessible via a domain in the form of : http://www.domain.com

After being deployed using Vercel.

**5.** **User Documentation**

**User Guide:**

**For users the website will be available at :** [**MediChat Homepage**](https://medi-chat-ai.vercel.app/)

**Homepage:**

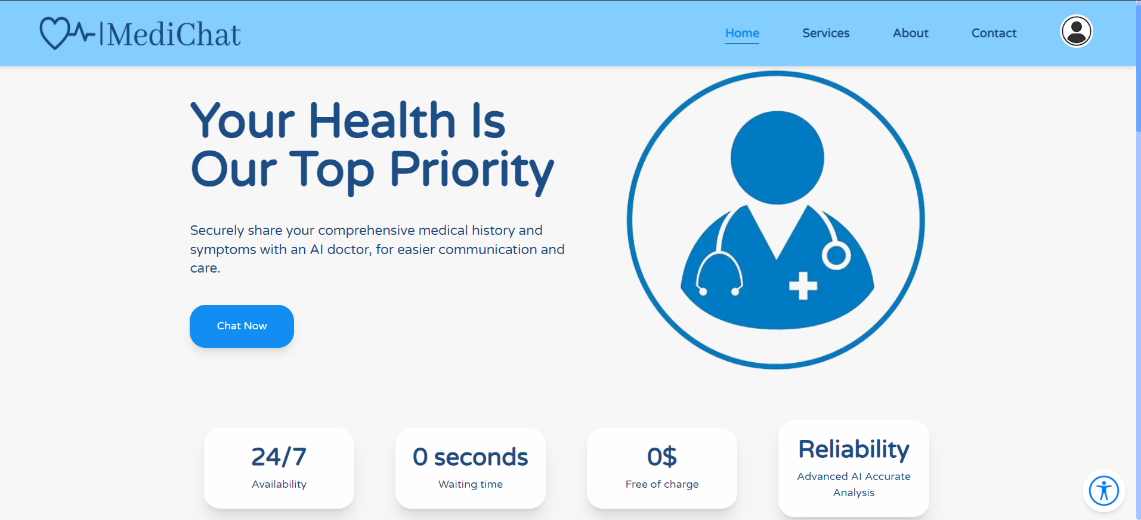
Welcome to MediChat, your go-to platform for AI-based medical consultations. On our homepage, you'll find general information about our service and an easy way to get started.

* Start Chatting: To begin a consultation with our AI Consultant, simply click the "Chat Now" button.
* User Registration/Login: You can access your account or register as a new user via the user button at the top of the page.
* Accessibility Features: For a more personalized browsing experience, use the accessibility button to increase font size and toggle dark mode.

Start Chatting by clicking this button

Access Profile or login/register

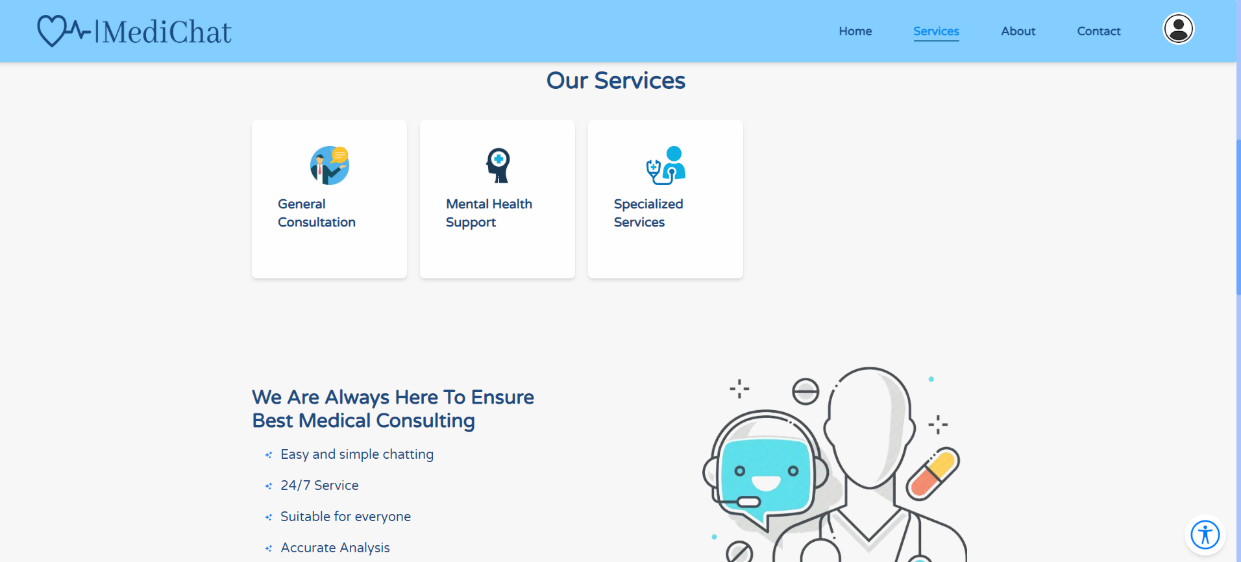
Fast scrolling to information sections



Accessibility Button

**Services Section:**

Services Section: Explore the wide range of AI-powered medical consultation services we offer, tailored to meet your healthcare needs efficiently and securely.



Click on the card will open additional information

**About us Section:**

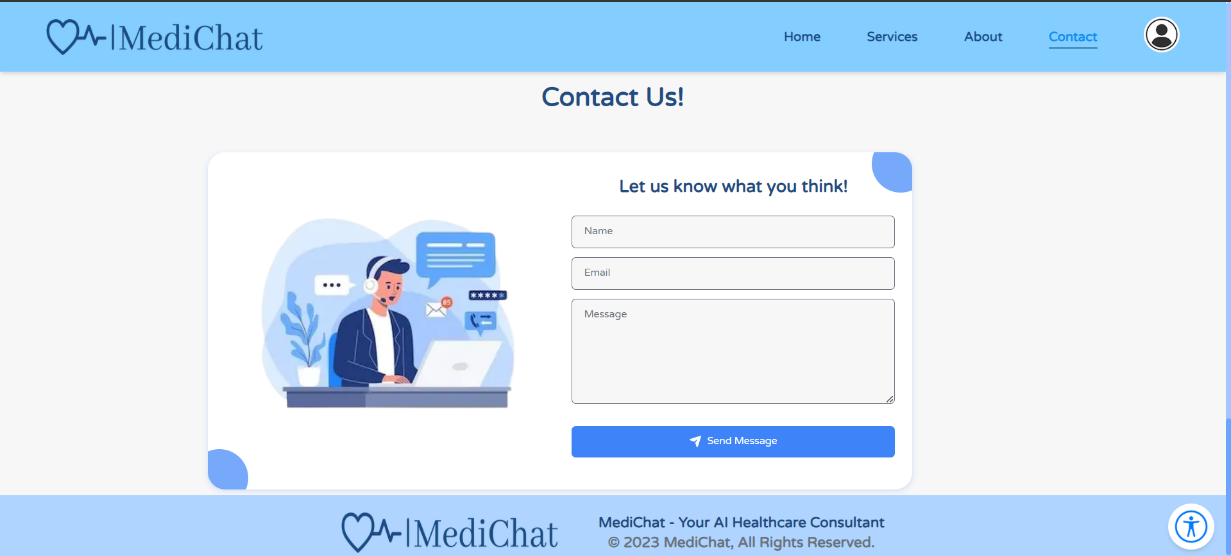
Learn more about MediChat, our mission to revolutionize healthcare through AI, and how we're making expert advice more accessible to everyone.

**תמונה שמכילה טקסט, צילום מסך

התיאור נוצר באופן אוטומטי**

**Contact Section:**

Have questions? Get in touch with us through our contact form, email, or phone, and we'll be happy to assist you.

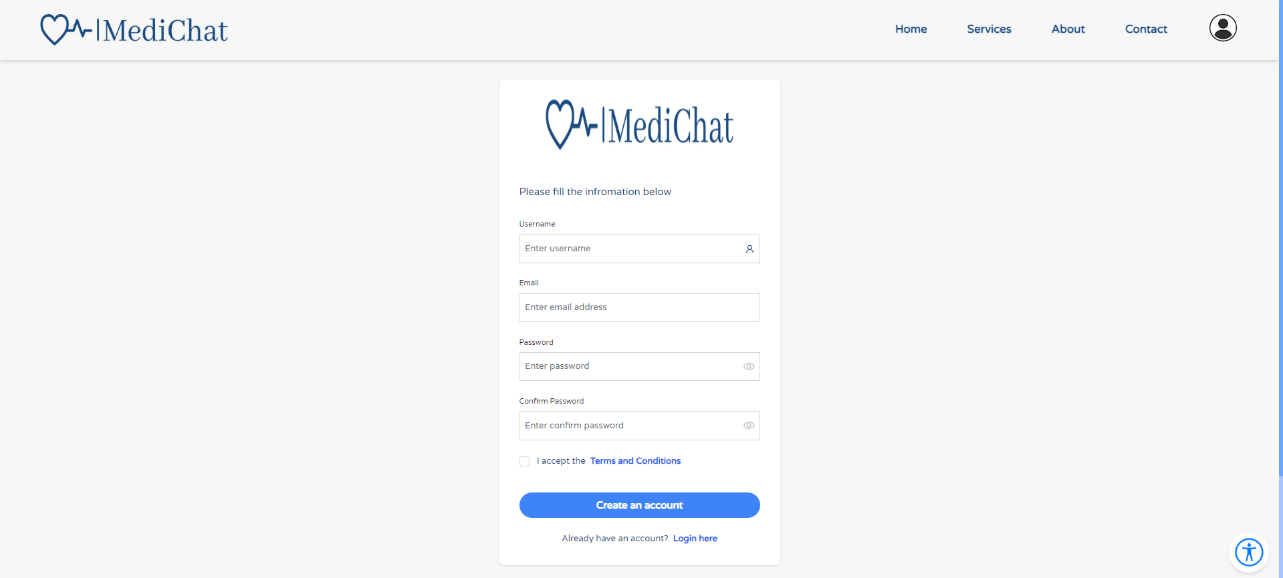


Fill this information to send us feedback

Send the feedback

**Registration Page:**

Create your MediChat account to access personalized consultations, save your medical history, and get the most out of our services.



Fill the information to register to the website

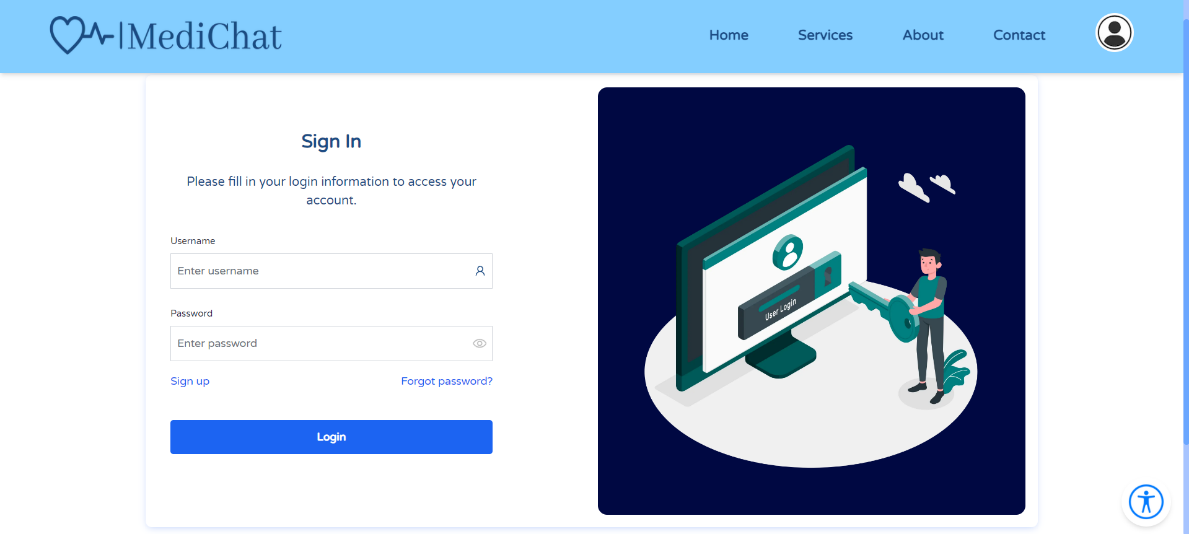
Accept the terms and condition

Create your account

**Login Page:**

Log in to your MediChat account to continue your consultations, manage your profile, and review past sessions.

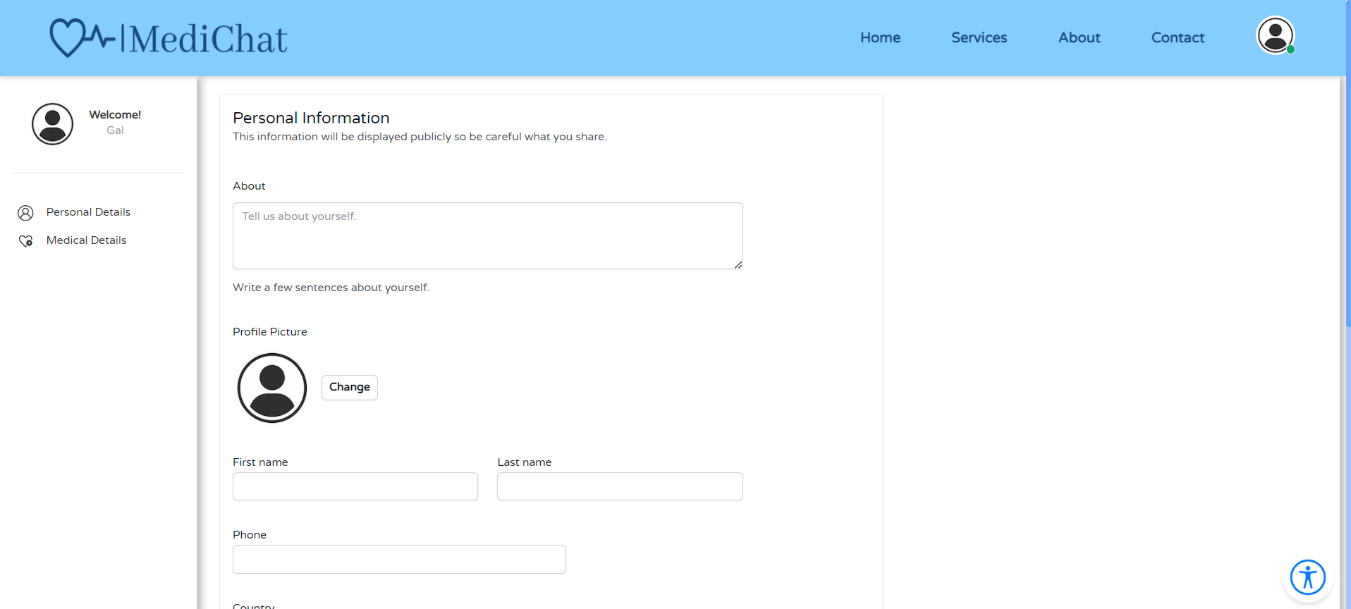
Enter Username and Password to login



Press to log in

**Personal Profile Page:**

Manage your personal information, update account details, and customize your preferences in your user profile.



**Medical Profile Page:**

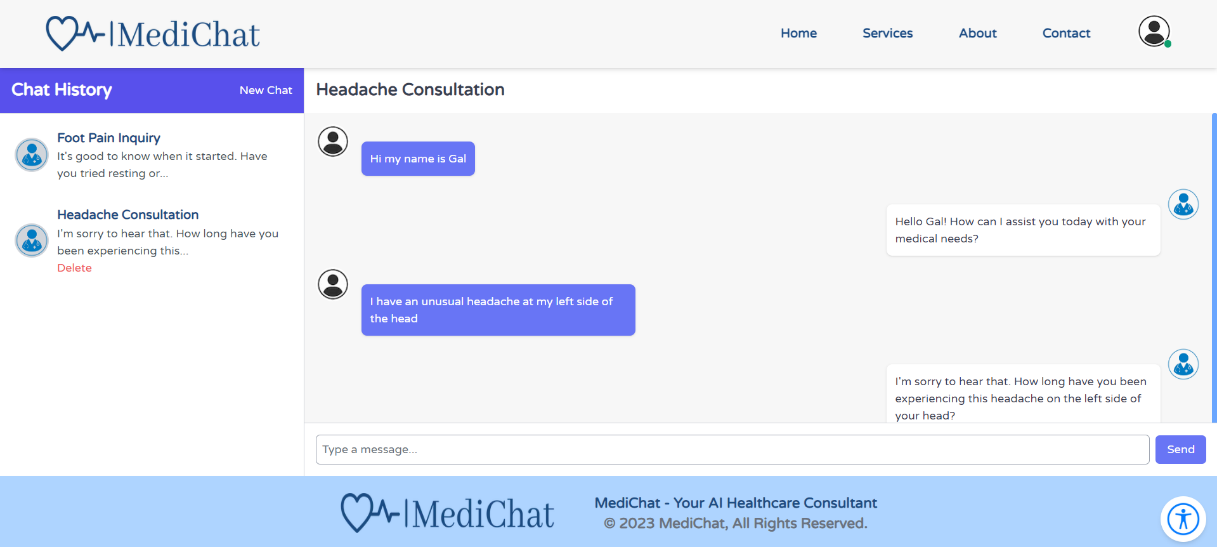
Keep track of your medical history, allergies, conditions, and past consultations to ensure personalized and accurate recommendations.

**תמונה שמכילה טקסט, צילום מסך, מספר, גופן

התיאור נוצר באופן אוטומטי**

**Chat Page:**

Engage in real-time conversations with our AI Consultant to receive instant medical advice tailored to your symptoms or questions.



Access chat history

Type your message

Press "Send" or Enter key to send the message

Delete the chat

**6.** **Verification and Evaluation**

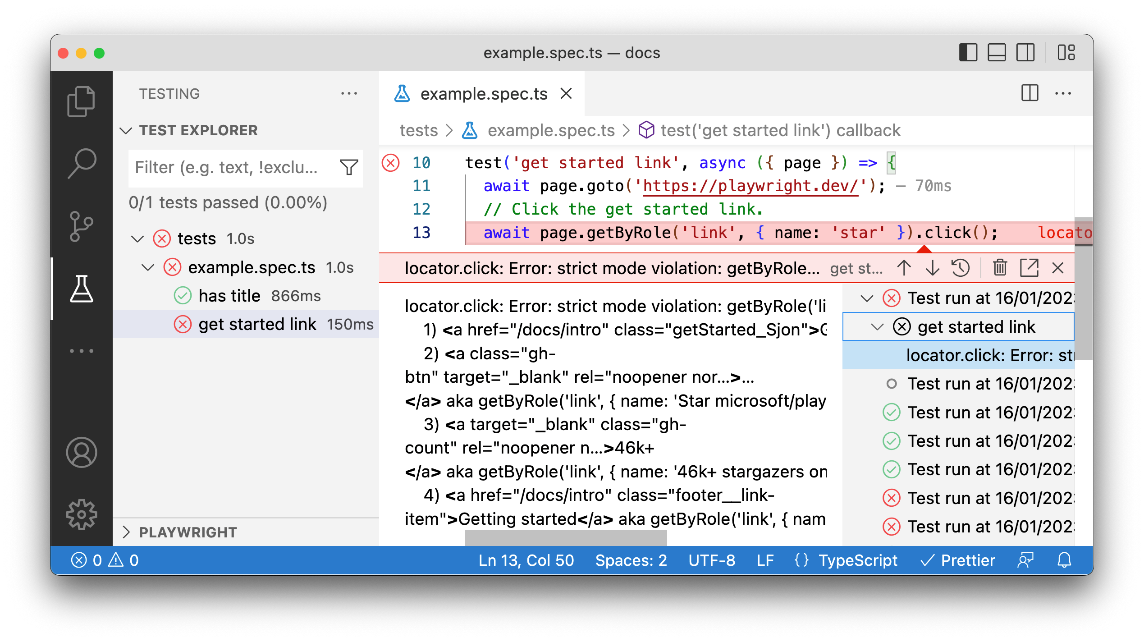
**6.1** **Testing**

Our testing strategy will be divided into two distinct components:

* Frontend Testing
* Backend\API Testing

For the frontend testing of our application, we have chosen Playwright, a testing tool known for its ease of use and robust testing capabilities. Playwright excels in running automated tests written in JavaScript, allowing us to simulate real user interactions on web pages. The framework supports end-to-end testing as well as both integration and unit tests, making it an ideal choice for ensuring our user interfaces are intuitive and perform consistently across all major browsers. Its built-in waiting mechanism and ability to handle asynchronous operations simplify the testing process, helping us to quickly identify and resolve UI-related issues.

Playwright Dashboard Example :



On the backend, we will employ Playwright to ensure the reliability and functionality of our APIs. Playwright is favored for its robust and versatile approach to end-to-end testing, allowing us to simulate real-world scenarios across multiple browsers with ease. With its powerful capabilities for automating complex workflows, Playwright enables us to comprehensively test our backend components, validating data integrity, server responses, and API handling capacities to meet both our standards and those required for modern web applications. By integrating Playwright into our continuous integration and deployment pipeline, we can ensure consistent and thorough testing, enhancing the overall quality and performance of our APIs.

**6.2** **Test Plan**

**Frontend**

* Validate user registration process.
* Validate user login functionality.
* Validate profile management features such as editing personal details and health metrics.
* Validate chat functionalities including starting new chats, continuing/viewing existing chat.
* Validate accessibility features including zoom in and dark-mode.

**Backend**

* Validate API endpoints for each process such as /api/login, /api/register, etc..
* Validate API requests / responses of AI technology being used
* Validate interpreter requests/responses and modifications

**6.3** **Evaluation**

We can evaluate our product by sending out a survey to various users to gather their opinions on the user interface and responsiveness. We'll also ask doctors and healthcare workers to assess the accuracy of our product.

Another option is to Utilize web analytics tools to track user interactions, session durations, and user demographics to gain quantitative insights into user behavior and engagement.

**Evaluation Metrics :**

**Responsiveness**: Measure page load times, rendering times, and responsiveness across different devices and browsers.

**Accuracy**: Measure the accuracy of diagnoses provided by the AI-driven doctor interface by comparing them with existing medical guidelines.

**User Satisfaction**: Gather user satisfaction ratings and feedback through surveys to understand overall satisfaction levels and areas for improvement.

1. **Maintenance Guide**

Our version for this product is beta version and not for production because

this is a health-related project that needs to cover lots of criteria and regulations to publish our project to the public.

To maintain and update the code, we are using the following methods:

* **Bug Fixes, Version Control and Release Management:**

**With each new feature and each new code addition, we might face conflicts, bugs and issues, in order to solve this, we decided to maintain the code using GitHub for version control and issues tracking**

* **Performance Optimization**

**We will conduct continuous testing and monitoring to find and optimize our project with Playwright, to locate regressions and monitor performance.**

**7.1 Developers & Contributes:**

In order to contribute to this project please do the following :

* Setting up the environment :
  + Clone the project using GitHub
  + Create a .env.local file in the main folder

MONGODB\_URI=mongodb+srv://<username>:<password>@cluster0.mongodb.net/<dbname>?retryWrites=true&w=majority

JWT\_SECRET=Your JWT Secret

REFRESH\_SECRET=Your Refresh JWT Secret

EMAIL=An email for contacting costumers {Gmail only currently}

EMAIL\_PASSWORD=email password

OPENAI\_API\_KEY=ChatGPT API KEY

BASEURL= Base URL of the application (localhost:8080/api/ for example)

GOOGLE\_TYPE=FROM GOOGLE SERVICE ACCOUNT JSON CREDENTIALS FILE

GOOGLE\_PROJECT\_ID=FROM GOOGLE SERVICE ACCOUNT JSON CREDENTIALS FILE

GOOGLE\_PRIVATE\_KEY\_ID=FROM GOOGLE SERVICE ACCOUNT JSON CREDENTIALS FILE

GOOGLE\_PRIVATE\_KEY=FROM GOOGLE SERVICE ACCOUNT JSON CREDENTIALS FILE

GOOGLE\_CLIENT\_EMAIL=FROM GOOGLE SERVICE ACCOUNT JSON CREDENTIALS FILE

GOOGLE\_CLIENT\_ID=FROM GOOGLE SERVICE ACCOUNT JSON CREDENTIALS FILE

GOOGLE\_AUTH\_URI=FROM GOOGLE SERVICE ACCOUNT JSON CREDENTIALS FILE

GOOGLE\_TOKEN\_URI=FROM GOOGLE SERVICE ACCOUNT JSON CREDENTIALS FILE

GOOGLE\_AUTH\_PROVIDER\_CERT\_URL=FROM GOOGLE SERVICE ACCOUNT JSON CREDENTIALS FILE

GOOGLE\_CLIENT\_CERT\_URL=FROM GOOGLE SERVICE ACCOUNT JSON CREDENTIALS FILE

**Example google service accounts credentials file**

{

"type": "service\_account",

"project\_id": "medichatai",

"private\_key\_id": "private key",

"private\_key": "-----BEGIN PRIVATE KEY-----\RSA KEY\n-----END PRIVATE KEY-----",

"client\_email": "serviceaccount@mail.iam.gserviceaccount.com",

"client\_id": "client\_id",

"auth\_uri": "https://accounts.google.com/o/oauth2/auth",

"token\_uri": "https://oauth2.googleapis.com/token",

"auth\_provider\_x509\_cert\_url": "https://www.googleapis.com/oauth2/v1/certs",

"client\_x509\_cert\_url": "https://www.googleapis.com/robot/v1/metadata/x509/~",

"universe\_domain": "googleapis.com"

}

* In the project folder open the terminal and run :

npm install { install all project dependencies }

Npm run {script name}

"test": "npx playwright test", for testing the api environment

"test-ui": "npx playwright test --ui", for testing the ui interaction

"dev": "next dev", for deploying the application locally

**Project Hierarchy :**

**- assets/ { assets folder used to store our images/logos and our CSS files }**

**- Logos/**

**- Images/**

**- index.css**

**- theme.css**

**- toastify-custom.css**

**- backend/ { the backend logic of the project }**

**- api-lib/ { API connections, mongo-db, etc.. }**

**- controllers/ { API controllers, handling requests from UI interactions }**

**- middleware/ { middleware for our API, interpreter logic before sending requests to ChatGPT }**

**- models/ { database models, users, chats, etc..}**

**- services/ { API services, the backend logic, communications with db and sending responses to controllers }**

**- pages/ { UI router, each page acts as an endpoint /index.html for example }**

**- [username]/**

**- api/ { api related routes }**

**- reset-password/**

**- \_app.jsx**

**- \_document.jsx**

**- 404.js**

**- chat.jsx**

**- index.jsx**

**- login.jsx**

**- register.jsx**

**- src/**

**- components/ { common UI components, shared variables, hooks, layout of the web, etc.. }**

**- page-components/ { logic of the UI pages, each page has a folder }**

**- tests/ { our playwright main folder for tests, api tests for backend and e2e ( end to end ) tests for UI }**

**- api/**

**- e2e/**

**After contributing to our project, open a PR on github and one of the maintainers will review it and comment on it, all PR's are only to the Staging branch.**

**8.** **References**

[1] Li, J., Dada, A., Puladi, B., Kleesiek, J., & Egger, J. (2024). ChatGPT in healthcare: a taxonomy and systematic review. *Computer Methods and Programs in Biomedicine*, 108013.

[2] Paredes, C. M. G., Machuca, C., & Claudio, Y. M. S. (2023). ChatGPT API: Brief overview and integration in Software Development. *International Journal of Engineering Insights*, *1*(1), 25-29.

[3] Wailthare, S., Gaikwad, T., Khadse, K., & Dubey, P. (2018). Artificial intelligence based chat-bot. *Artificial Intelligence*, *5*(03), 2305-2306.

[4] Qian, H., Dong, B., Yuan, J. J., Yin, F., Wang, Z., Wang, H. N., ... & Ning, B. T. (2021). Pre-consultation system based on the artificial intelligence has a better diagnostic performance than the physicians in the outpatient Department of pediatrics. *Frontiers in Medicine*, *8*, 695185.

[5] Cançado, L. Vera, Vendramine F. M. Corrêa, A. Dalila, Oliveira, J. Elizângela, and Castro. P. S. Dagmar. (2017) “Revisiting the Four Faces of Human Resource Management.” In: ANPAD Meetings – Enanpad, 2017. São Paulo/SP – 1-4. [3] Nascimento, M. Alexandre, and Queiroz. M.C Anna. (2017) “Overview of research on Artificial Intelligence in Administration in Brazil.” In: ANPAD Meetings – Enanpad, 2017. São Paulo/SP – 1-4.

[6] Loebbecke, Claudia, and Picot, Arnold. (2015) “Reflections on societal and business model transformation arising from digitization and big data analytics: A research agenda.” The Journal of Strategic Information Systems, 24 (3): 149-157. doi: 10.1016/j.jsis.2015.08.002.

[7] Morley, J., Machado, C. C., Burr, C., Cowls, J., Joshi, I., Taddeo, M., & Floridi, L. (2020). The ethics of AI in healthcare: a mapping review. *Social Science & Medicine*, *260*, 113172.

[8]Meng, M., Steinhardt, S., & Schubert, A. (2018). Application programming interface documentation: What do software developers want?. *Journal of Technical Writing and Communication*, *48*(3), 295-330.

[9] Ali, O., Abdelbaki, W., Shrestha, A., Elbasi, E., Alryalat, M. A. A., & Dwivedi, Y. K. (2023). A systematic literature review of artificial intelligence in the healthcare sector: Benefits, challenges, methodologies, and functionalities. *Journal of Innovation & Knowledge*, *8*(1), 100333.‏

[10] ADA - <https://ada.com/>

[11] Your.MD - <https://yourmd.online/>

[12] Bajwa, J., Munir, U., Nori, A., & Williams, B. (2021). Artificial intelligence in healthcare: transforming the practice of medicine. *Future healthcare journal*, *8*(2), e188-e194.‏

[13] Levesque, J. F., Harris, M. F., & Russell, G. (2013). Patient-centred access to health care: conceptualising access at the interface of health systems and populations. International journal for equity in health, 12, 1-9.‏

[14] Laranjo, L., Dunn, A. G., Tong, H. L., Kocaballi, A. B., Chen, J., Bashir, R., ... & Coiera, E. (2018). Conversational agents in healthcare: a systematic review. Journal of the American Medical Informatics Association, 25(9), 1248-1258.‏

[15] Bouwman, H., Nikou, S., Molina-Castillo, F. J., & de Reuver, M. (2018). The impact of digitalization on business models. Digital Policy, Regulation and Governance, 20(2), 105-124.‏

[16] TidalHealth Peninsula Regional -https://en.wikipedia.org/wiki/TidalHealth\_Peninsula\_Regional

[17] Portal Telemedicina - <https://cloud.google.com/customers/portal-telemedicina-gcp>

[18] Amsterdam UMC - <https://www.amsterdamumc.org/en/organization/amsterdam-umc.htm>

[19]  Vercel. (2023). Deploying to Vercel. Vercel Documentation. <https://vercel.com/docs/deployments/overview>