Academia de Studii Economice din Bucureşti
Facultatea de Cibernetică, Statistică şi Informatică Economică
Specializarea Informatică Economică / Cibernetică Economică

Web application for medical clinic

Bachelor's Thesis

Coordonator
Prof. univ. Cotfas Liviu-Adrian

Absolvent
Serban Cristi

București 2023

Declaration on the originality of the content and assumption of responsibility

I hereby declare that the results presented in this paper are entirely the result of my own creation, except where reference is made to the results of other authors. I confirm that any material used from other sources (journals, books, articles, and websites) is clearly referenced in the paper and is indicated in the list of bibliographic references.

CONTENT

1. Introduction

In the outstanding article "*The Importance of Maintaining Health and Functioning: The Perspective of Older People Receiving Home Care Services" by Vass et al. (2012) [1],* the author states noteworthy reasoning on why health has an essential role in modern times. The very first reason why health is salient is being able to maintain independence and the wherewithal to perform day-by-day tasks. Furthermore, quality health is requisite for the overall quality of life. Being healthy empowers individuals to have the benefit of enjoying their life and pursuing activities they find fulfilling. Last, but not least, another foremost justification is that good health may prevent or manage chronic conditions. These chronic conditions can be consequential to an individual's quality of life.

In the present, it is given the impression that the number of web applications for medical clinics has seen quite a considerable growth. This growth may be chalked up to various factors, including the exploitation of technology in healthcare, the uptrend in demand for approachable healthcare services, and the necessity to enhance patient outcomes while reducing costs. Another component guiding the inevitable growth of web applications in healthcare is the prerequisite to streamlining administrative tasks and diminish paperwork. By digitalizing medical records and appointment scheduling, healthcare suppliers are able to save time and resources, empowering them to focus on patient care. Nonetheless, the expanding adoption of web applications in healthcare is metamorphosing the way people retrieve and receive medical care, making it more organized, accessible, and patient-centered.

It goes without saying that there are some benefits that, from my humble point of view, are pretty paramount when making use of this kind of application, such as: being able to see the list of doctors with their available time when searching for a certain examination; this is to a great degree for patients who have a living in rural areas or are at considerable distance; having a record of not only previous but also forthcoming appointments available at all times is at best to the patient’s advantage as accurate and up-to-date information. It goes without saying that this will diminish errors, boost communication between healthcare providers, and, in the long run, improve patient care.

However, over and above that, even if we are swiftly shifting into a more modernized era, there are still a lot of people who are not friends with technology and, let’s say, who still prefer to make a phone call for an appointment. Nevertheless, it is important to bear in mind, that in this day and age, being technologically literate is becoming increasingly essential for accessing services and resources. Therefore, even if less tech-savvy individuals may in the first instance struggle with using a healthcare web application, I would affirm that it is quite important for them to make an effort, absorb knowledge, and learn how to use it. This may imply seeking help from family members, friends, or health providers. But, in spite of that, by investing time and endeavor into learning how to use these tools, individuals can better be in charge of their health, access care more skilfully, and refine their quality of life.

Notwithstanding, the present technology comes with the extension of present threats. As healthcare web applications continue to proliferate, it is of substantial significance to recognize the importance of data privacy and security. These applications operate on vast amounts of personal and sensitive information, including medical histories, diagnoses, and treatments. Any infringement of this data may have appalling consequences for patients, and we could speak about identity theft, discrimination, and even compromised healthcare.

In the wake of contemporary high-profile data breaches in the healthcare sector, patients have become anxious about the assurance of their personal data. Those responsible must counter these concerns by ensuring that the web applications are secure and patients privacy is protected at all times; *Healthcare Data Breaches: Insights and Implications [2]* article would support all the above mentioned.

Chapter 2. Web applications for medical clinics

2.1 The objective of the project

A tedious and unexciting waiting hall, the long queues, the hopelessness of getting inside the doctor's chamber, the depressing faces of ailing patients— this would be the representation one would imagine maybe a decade ago while paying a visit to a medical centre. However, in recent times, this plot has been exceptionally altered thanks to the web development industry's flourishing healthcare apps.

Furthermore, the study aims to explore how healthcare apps can contribute to reducing diagnostic errors by providing a centralized and accurate record of patients' health information, which can aid healthcare providers in making informed decisions. Additionally, the project will investigate the user experience of healthcare apps, including factors that may impact user adoption and engagement, and potential biases in user reviews. Through a comprehensive examination of these aspects, the research aims to provide insights into the potential benefits and limitations of utilizing healthcare apps in improving the efficiency and accessibility of healthcare services.

Nevertheless, minimizing the liability of misdiagnosis is also a paramount objective in healthcare, and such applications are able to have a critical role in achieving this intention. Managing patients' records manually is sensitive to diagnostic blunders, inaccuracies and is time-wasting. However, the health web apps invalidate all such potential threats that might prove fatal for the long-suffering individuals. It is unquestionable, that maintaining a meticulous record of a patient's health condition can aid doctors in establishing the correct medicine with the correct dosage and chemical compositions. The article "*Global Misdiagnosis insides - medical error statistics by countries*" [3] only strengthens my point of view that, between diverse components that lead to misdiagnosis, system flows have a salient word to say: problems with connection or cooperativeness of care and handiness of medical record data. Delving into more quantitative specifics, the disruption amidst the patient and the medic during the clinical encounter seems to be the very first reason (79%) of why this happens.

In addition, the project will also consider potential challenges such as data privacy and security concerns, barriers to access for certain populations, and potential limitations in the accuracy and reliability of information provided by healthcare apps. By addressing these potential limitations, the research aims to provide a holistic and balanced perspective on the impact of healthcare apps on improving the efficiency of accessing healthcare services.

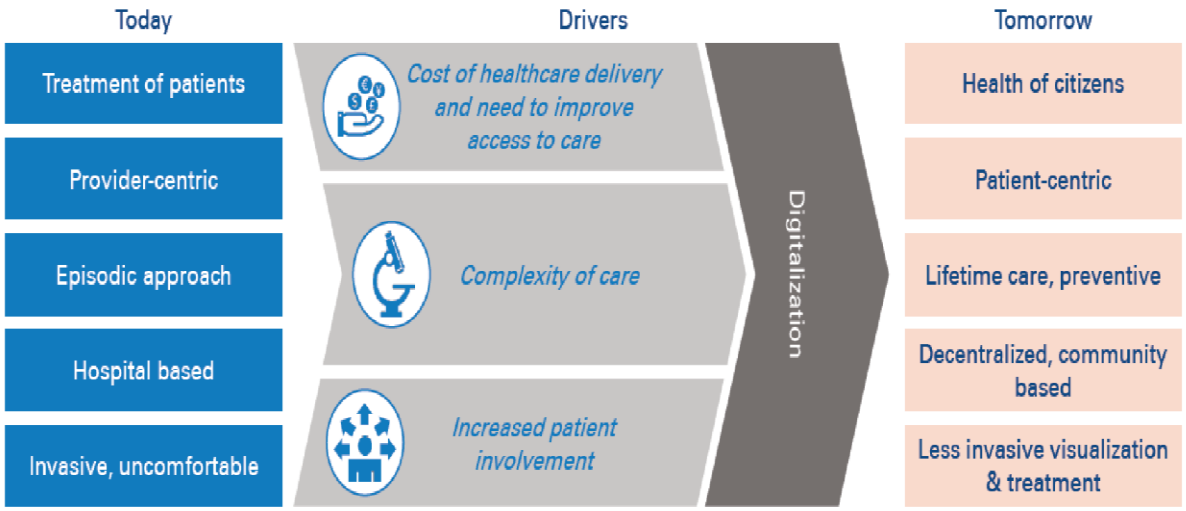

Figure 1 - Concepts

Source: (https://www.adlittle.com/no-en/insights/report/hospital-information-systems-digitally-enabled-era)

This image Figure 1 - Concepts would synthesize in a highly straightforward manner the most meaningful concepts I had the intention to be handled when I first thought about this topic. It is in clear view how the digitalization can only improve our living.

2.2 Application framework in web environment

In the immediate present, the number of existing frameworks on which you can build a web app has seen an uninterrupted expansion, and this can be stated for both backend and frontend technologies. Despite this vast diversity, my preferences were React.js for the UI/UX segment, ASP.NET for the backend component, and SQL Server for the database server. The IDEs used are pretty commonplace, in order for every technology mentioned just above: Visual Studio Code, Visual Studio, and SSMS.

Bringing up an overview of the application, I can avow that my desire is to invent a straightforward application when speaking about the graphical user interface part, with the argument that this sort of application is designed to be used by the elders as well, not just by teenagers or, in general, by tech-skilled people. Furthermore, adding some intriguing functionalities would also be a part of my design.

Engaging in more detailed information, I found that the experience of this kind of web app should be nearly established on a personal account created on an email that has an obligation to abide by the requirements of the industry, meaning that it should respect the pattern...@yahoo.com or...@gmail.com, etc. depending on the email provider that the patient utilizes. What's more, this comes in conjunction with a strong, secure password that needs to be long enough and have at least one special character. Thenceforth, after registering and logging in on the site, a home page will be uncovered, and the functionalities will be revealed to the user: ask for an examination, see the made appointments, read the reviews of the clinic,

consult the price list in order to be unconditionally aware and download it as an Excel file, and ask for forthcoming services wanted. Finally, the final part of the menu is the FAQ.

Nonetheless, the local storage of the database also plays a significant role in storing various information, such as the registry of doctors with their daily availability, the price list, the reviews, and the created accounts, among others. However, I am of the opinion that special attention should be given to sensitive information. Therefore, it is imperative to store passwords in a secure manner, using hash functions and not in plain text; this comes along with the just-mentioned strong, secure password just above.

Additionally, for the local storage of the database, I tried to introduce a slight usage of what is nowadays called cloud computing. My pick, having in mind that I used technologies such as SQL Server and ASP.NET, was indubitably Microsoft Azure. As a very succinct description, Microsoft Azure is a cloud computing platform that offers different services for creating applications globally. Taking this into account, the review part is in possession of a field that allows users to upload a photo. It is self-evident that the photo won't come to light on the page, and this feature is not a condition for the review to be posted.

2.3 Other such web applications

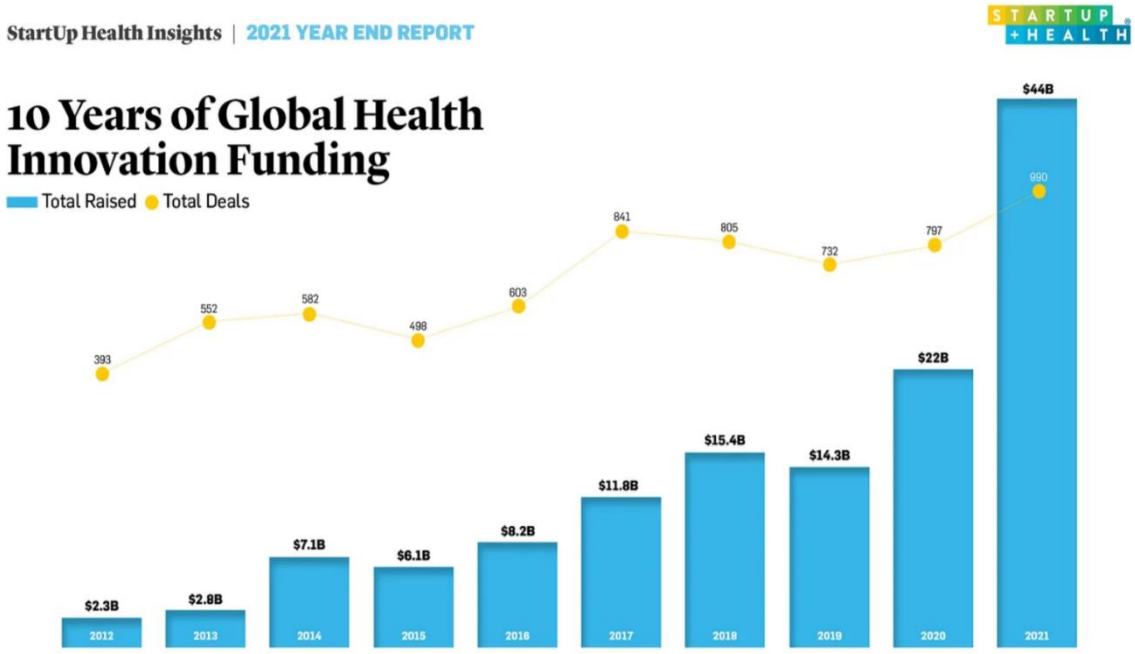

Figure 2 - Reports

Source: 2021 Year-End Insights Report (StartUp Health)

When looking at the chart above Figure 2 - Reports and assessing it with a closer look, we can agree with the very clear fact that from the starting point of the investigation, 2012, and up to the last year taken into account, 2021, the digital health area of interest has established tremendous growth. According to the "2021 Year-End Insights Report: $44B Raised Globally in Health Innovation, Doubling Year Over Year" [4] in 2011, just about $2B was deposited into this field. However, at the end of 2021, the dialogue will have undergone a complete shift, as a total of $44 billion has been allocated worldwide for funding health innovation. We are speaking about a 20x increase in just 10 years.

The exponential rise in digital health investments, which have doubled from one year to the next in 2020, can also be imputed to the well known COVID-19 pandemic, as announced by WHO early in the year. Furthermore, it is quite paramount to observe that the conclusion was not inevitable. From square one, investors became cautious, and markets experienced a deceleration in March and April 2020, with major media outlets hinting at a theoretic cooling in the digital health market.

Nevertheless, we ought to take into account also some national providers of such services.

- **Regina Maria**

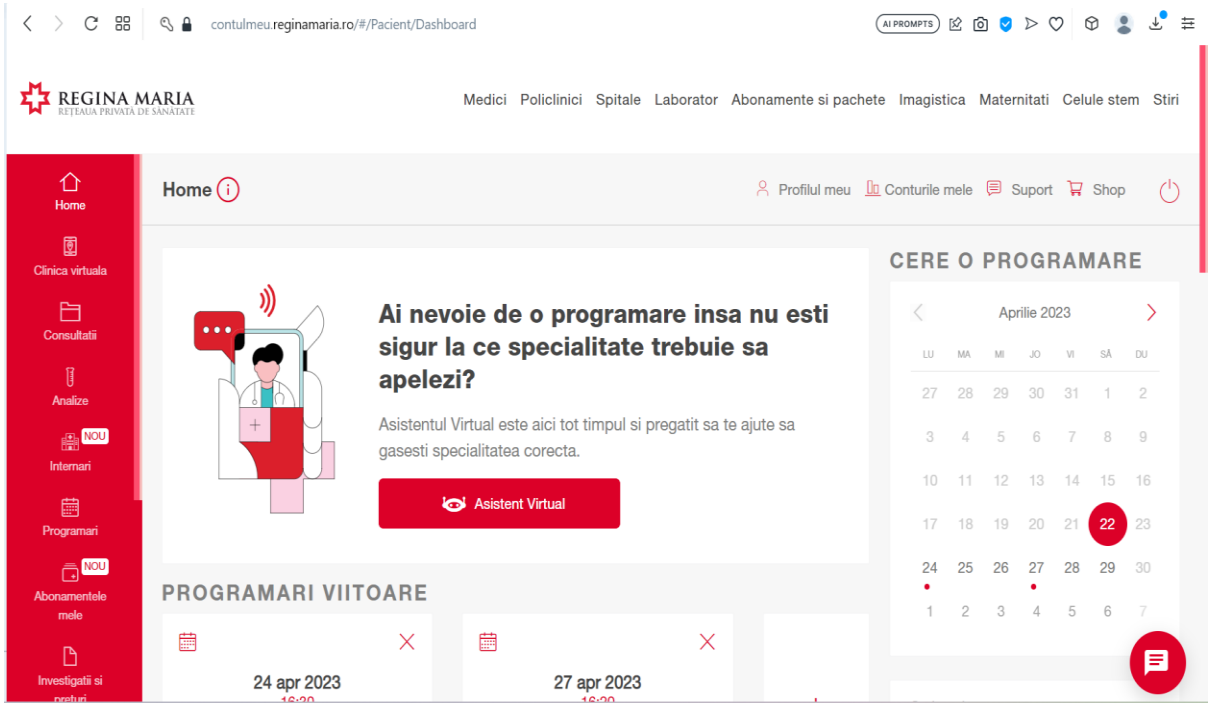

Figure 3 - Regina Maria Dashboard

Source: Regina Maria Web Application

Figure 3 - Regina Maria Dashboard; Stating just some words about the private health network, as how they describe themselves, it is one of the most essential supplier in this area of interest.

As it can be effortlessly discernible, the Regina Maria web application provides a great amount of information to the patient on the left side: past and forthcoming consultations, blood

tests, subscriptions, and other things related. Their interface seems to be rather user-friendly, with all the features just one click away. For example, requesting an appointment would take just a few clicks in the immense, impossible-to-miss "Cere o programare" right part of the page.

According to the web site, they place a great deal of importance on some features: being able to schedule an appointment no matter what the hour is, with the details left to be handpicked by the patient, such as the desired doctor, at the adequate hour, at the nearest accessible location. What's more, it seems like they also bring to the forefront that the application will always advise the first time slot available. This list of characteristics comes to an end with the capability of checking the price list before the actual consultation.

They also have a section with a palette of subscriptions divided into two sections: the ones for ordinary individuals and the ones for companies. Those who are built for individuals come, obviously, in a limited number, and to be more exact, 3, starting with a price of 74 RON per month and ending up with 214 RON per month. The contrast in price resulting from the features that a subscription has. The very same reasoning is the same for the companies subscriptions.

All in all, The Regina Maria web application offers a wealth of information to patients, including past and upcoming consultations, blood tests, and subscriptions. The interface is user-friendly, with easy access to features such as appointment scheduling. The application allows patients to handpick appointment details such as the desired doctor, time, and location, and advises the first available time slot. The web site also offers subscription options for individuals and companies, with varying prices based on the features included.

- **MedLife**

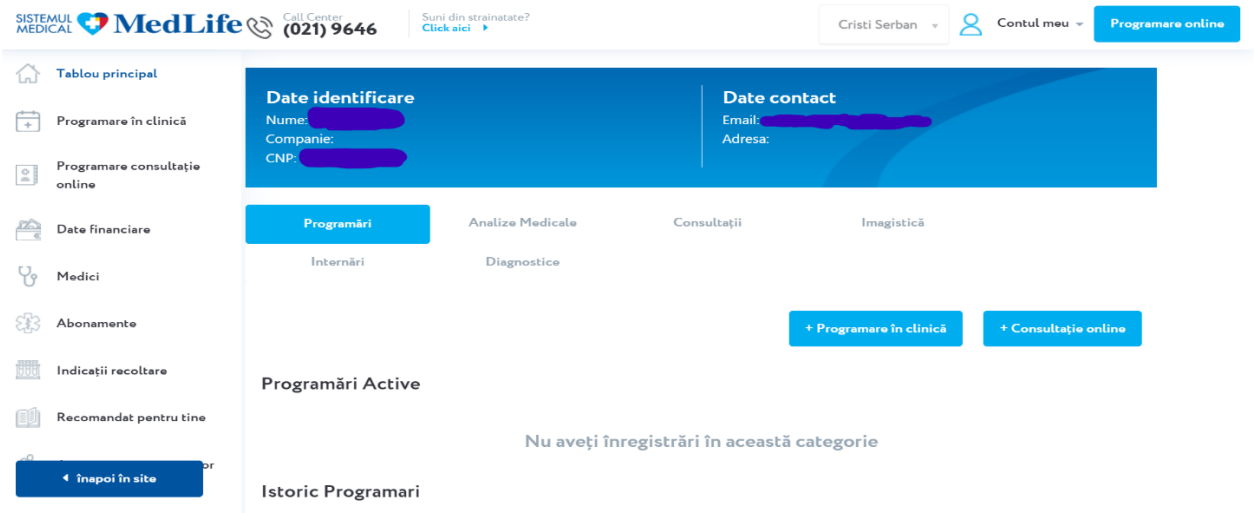

Figure 4 - MedLife Web App Dashboard

Source: MedLife Web Application

Another company that provides health services in Romania is MedLife. Furthermore, on a simple Google search, they portray themselves as a medical system. (Figure 4 - MedLife Web App Dashboard)

Delving further into the specifics about what truly interests me, the web app, it is more or less evident that MedLife's interface is similar to Regina Maria's one. Based on the very same pattern, the menu is also located on the left part of the page, with the header containing some data about the owner of the account. Nearly equivalently, there is a button for asking for a consultation. Moreover, the description of the app as a whole is quite identical to the competitor's.

As things weren't quite alike enough, Medlife also comes up with their very own catalog of subscriptions, the imbalance being made by the tiniest detail, indubitably not taking into account the pricing: they have 4 types of subscription instead of 3.

All things considered, MedLife, a healthcare services provider in Romania, appears to have similarities in its web app interface with its competitor, Regina Maria. The layout, menu placement, and description of the app are nearly identical, with the only notable difference being the number of subscription types offered. Despite these similarities, pricing is not taken into account.

Chapter 3. Technologies used

3.1 Programming Languages used

Habitually, when making mention of a web application we are speaking about three foremost constituents: client-side, server-side and a database server. But in spite of that, in my case, the third one has an individual role, and that is to store data. For this very rationale, the emphasis will be on the client-side, respectively server-side languages, React.js and ASP.NET.

React.js is a JavaScript framework viable for building user interfaces for web applications. Developed by Facebook at the moment of writing Meta, React empowers the innovator to bring into being reusable UI components and manage the state of those components as they convert over moments in time. The admiration for this framework has arisen because of its user-friendly nature, flexible architecture, and high productivity. It is a widely adopted tool among web developers for crafting dynamic, interactive, and responsive user interfaces for web applications.

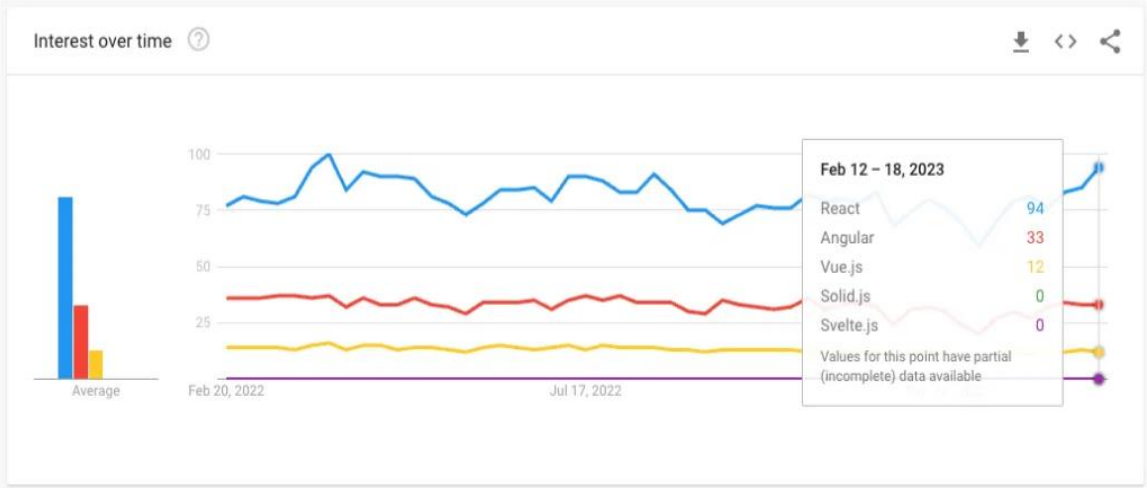

Figure 5 - Interest over JS frameworks

Source: Google Trends

The image above Figure 5 - Interest over JS frameworks reveals the interest over time in a mixed bag of JavaScript frameworks and how React.js occupies the first place. What's more, according to *lambdatest.com*, there are a noteworthy number of distinguished benefits when referring to this peculiar framework, and those would be: easy integration, direction dataflow, simpler syntax, virtual DOM, and the list goes on and on.

On the other side of the rationale, as mentioned at the start of the chapter, there are two central programming languages used to develop this web application, and with the discourse about React.js being accomplished, it is now ASP.NET's turn.

ASP.NET is a web framework developed by Microsoft that is open source and designed for creating modern web apps and services using .NET . It operates on top of the HTTP protocol and employs HTTP commands and procedures to enable collaboration and communication

between the browser and server. While there are other alternatives to ASP.NET, such as C#, Visual Basic.Net, JScript, and J#, ASP.NET has a long history of development and has been constantly evolving since version 1.0. The latest version used for this project is ASP.NET 6.0.

To assist developers in creating strong and scalable web applications, ASP.NET offers various capabilities, including but not limited to security, caching, session management, and authentication.

3.2 Visual Studio 2022

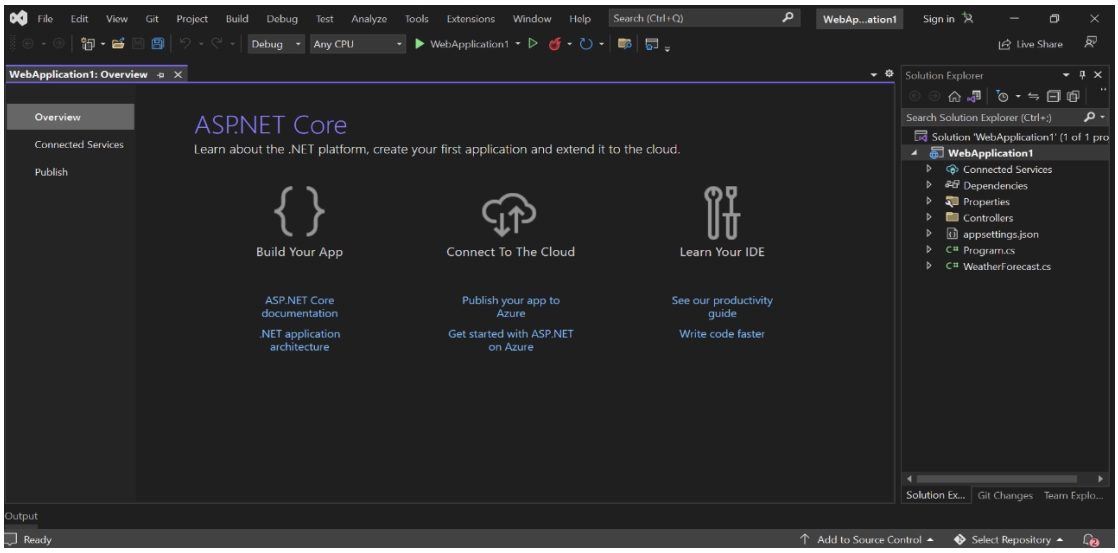

Figure 6 - Visual Studio 2022

Source: Visual Studio 2022 ASP.NET Web Core API project

At the very cornerstone of my back-end share of the web application is ASP.NET, as detailed by Microsoft as "*open source web framework" [5]*. For this consideration, my preferred IDE to further develop the project was indubitably Visual Studio 2022, which is also an "integrated development environment" also created by Microsoft. This platform has suffered different improvements through the years, the very first being Visual Studio 2013.

It goes without saying that Visual Studio 2022 delivers a significant amount of possibilities when referring to the options available for creating an application, starting with the indispensable debugging and ending up with advanced collaboration features such as Live Share, Azure DevOps integration; however, for the goal of the project, I will concisely characterize it as a whole.

Therefore, on the right side of the window is the solution explorer, which supplies the user with a handful of methods to instantaneously access the project: manage files, projects, and solutions. What's more, right next to the "Solution Explorer" stands the "Git Changes" section that empowers developers to manage version control when speaking about their projects and, eventually, collaborate with team members, making teamwork and project management way more efficient.

3.3 Visual Studio Code

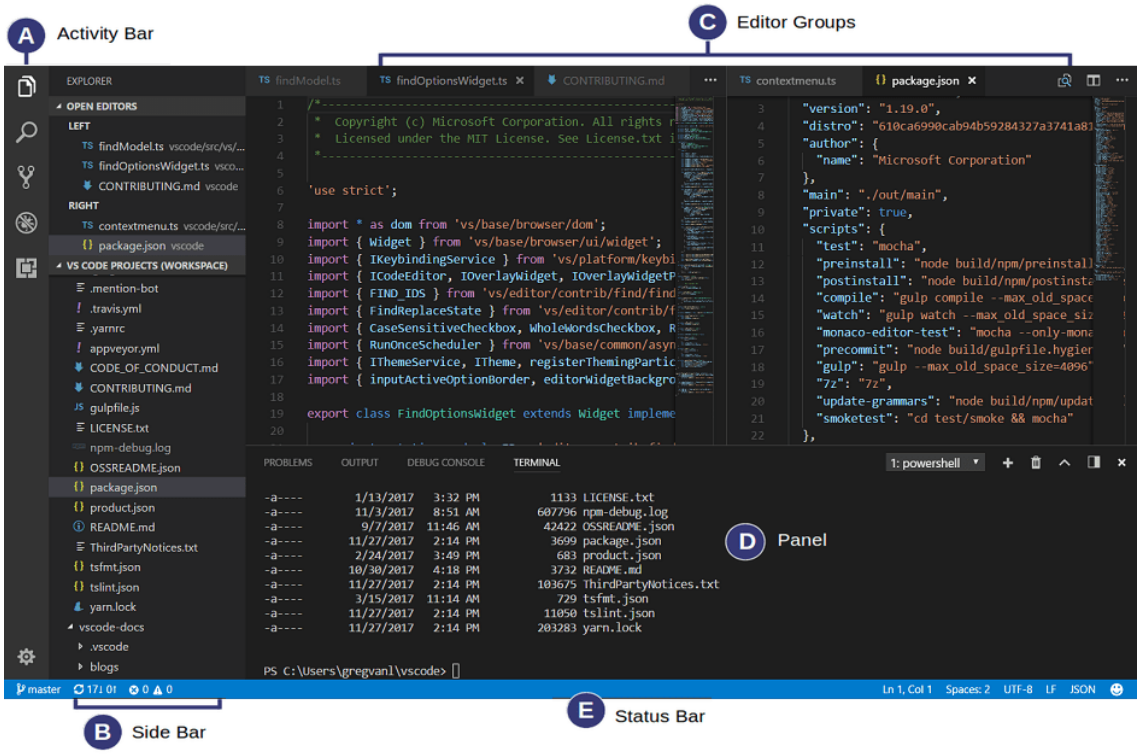

Figure 7 - Visual Studio Code

Source: Visual Studio Code Documentation (https://code.visualstudio.com/docs/getstarted/userinterface)

According to official documentation, Visual Studio Code (Figure 7 - Visual Studio Code) is "*source-code editor made by Microsoft*". *[6]* Therefore, this description could be supplemented. The list of features incorporates things such as syntax highlighting, debugging, pieces of code known as snippets, being able to change the general theme, installing extensions, and the list goes on and on. Nevertheless, as the very final piece of information, in the *Stack Overflow 20222 Developer Survey*, [7] this IDE was placed in the very first place when speaking about popularity.

Nevertheless, going into slightly significant details, in agreement with the Visual Studio Code Documentation, the user interface is a mixture of five predominant parts, as can be seen: Activity Bar to be able to change the views immediately, Side Bar with data about the project, Editor being the most important part for editing the files, Groups, Panel to debug, and Status Bar with reports about the project.

Thus, with this being presented, next I can introduce the most significant technologies I used when referring to the front end.

The responsiveness of the whole front-end architecture is centered on Bootstrap. Bootstrap is a chargeless, free software with a simple and accurate central aim: to simplify the creation of responsive websites by offering collections of templates and syntax design objectives.

Why ought to use Bootstrap? The foremost justification would be rather stimulating: the simplicity of usage. Because of its widespread usage, numerous online tutorials and forums exist to assist beginners in getting started. The upcoming argument is the responsive grid. The software has in its components a pre-built grid system, which erases the requirement to bring one into being. This system is built up of rows and columns that empower you to generate a grid within the primitive grid instead of incorporating media queries directly into the CSS file. Last but not least, Bootstrap documentation[8], which is quite intuitive and rather approachable, has a salient word to say nonetheless, and last but not least, the ease of use, that look like this:

Figure 8 - Bootstrap example 1

or this:

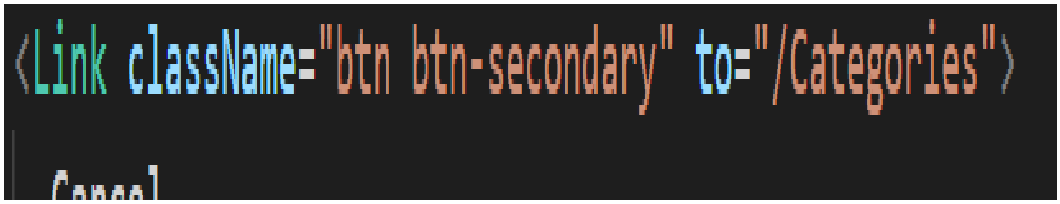

Figure 9 - Bootstrap example 2

As a conclusion, according to w3techs.com [9], out of all the websites whose JavaScript libraries we are aware of, Bootstrap is utilized by 25.9%, which accounts for 21.3% of all websites. This indicates that Bootstrap is a widely adopted and popular JavaScript framework among website developers.

Another paramount factor in my web site application is the validation of the forms. This, would be done with Yup. Yup is a schema constructor for analyzing and interpreting runtime values. It facilitates modelling complex, interconnected verification or value transformation and is succinct yet powerful. Yup is an outstanding modeler which embraces both server-side and client-side validation equally well and comes with built-in asynchronous verification features.

```
<Formik
initialValues={props.model}
onSubmit={props.onSubmit}
validationSchema={Yup.object({
    name: Yup.string().required('This field is required'),
    consultationDate: Yup.date().required('This field is required').nullable(),
```

Figure 10 - Technology used

All things considered, this IDE was my alternative to building the front-end part using React.js.

3.3 SQL Server and SQL Server Management Studio (SSMS)

SQL Server is a relational database management system created purely by the tech giant Microsoft, and expanded on SQL, a programming language commonly used for interacting with relational databases in a standardized manner. Therefore, as a database server, its most consequential purpose is to facilitate the storage and retrieval of data requested by other software applications, either on the same computer or a remote computer, utilizing the client-server module.

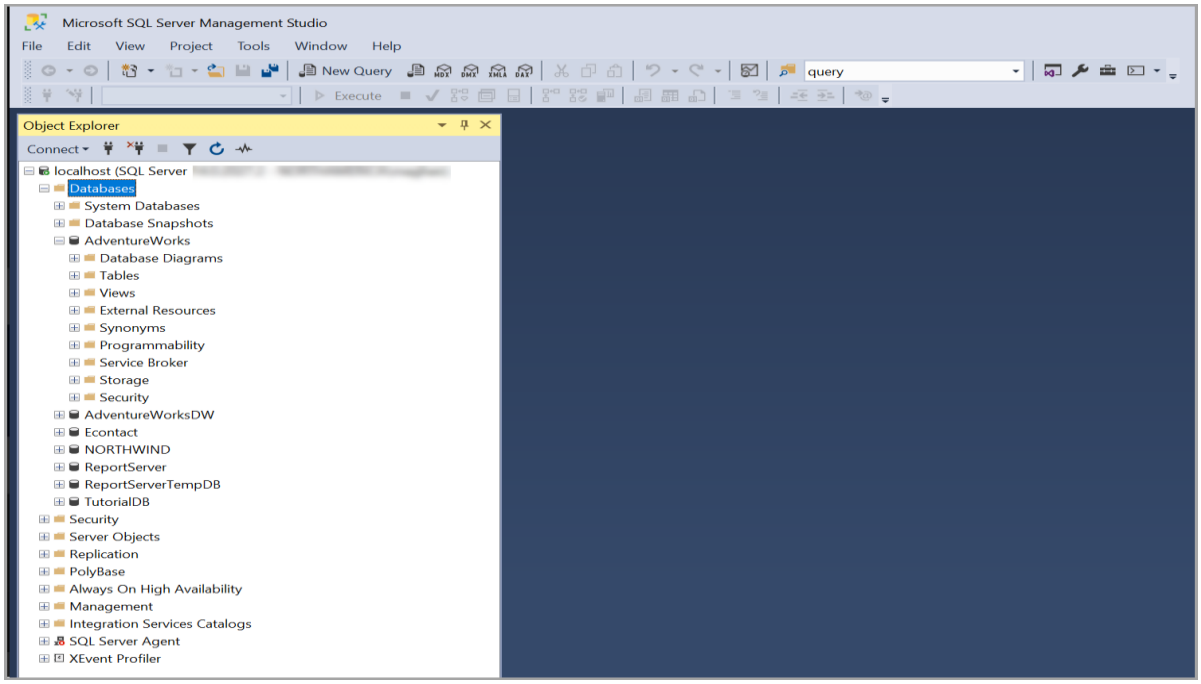

Figure 11 - SSMS

Source: SSMS Documentation (https://learn.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16)

In conformity with the documentation [10], SQL Server Management Studio (Figure 11 - SSMS) is an integrated tool that bestows a graphical user interface for conducting a considerable amount of aspects of SQL infrastructure: SQL Server[11], Azure SQL Database, Azure SQL Managed Instance, etc. Nonetheless, it provides a panoramic range of features and benefits that empower the ability to manage database objects, write and execute queries, and design server settings. In summary, SSMS proves to be a potent and instinctive tool for handling SQL infrastructure.

Making allowance for the things suggested above, I am of the opinion that my reasoning for electing this technology to compound the local data storage of my project is absolutely unambiguous: as with other technologies, it was also created by Microsoft.

On the other hand, a critical component that send data through the API into the database is Axios library. Axios is a widely recognized JavaScript [12] package aimed at making it less difficult to submit HTTP requests. It delivers a straightforward API with tools for controlling answers, sending HTTP queries, and manipulating data in numerous manners. Axios has proven itself as an outstanding choice for developers seeking for a trustworthy and efficient tool for managing HTTP communication in their applications due to its intuitive interface and impressive features like computerized JSON parsing and request cancellation.

```
await axios({
  method: "POST",
  url: urlCategories2,
  data: formData,
  headers: { "Content-Type": "multipart/form-data" },
});
history.push("/categories2");
} catch (error: any) {
if (error && error.response) {
  setErrors(error.response.data);
}
```

Figure 12 - Axios Technology

Data is able to be communicated to a server using the HTTP POST (Figure 12 - Axios Technology) request utilizing the versatile Axios POST mechanism. Developers may basically oversee the server's response and include information within the request body, such as form inputs or JSON payloads. This approach makes the process simpler to supply data to web applications and makes it feasible for the client and server to share information properly.

```
async function edit(entityToEdit: TCreation) {
    try {
        if(props.transformFormData) {
            const formData = props.transformFormData(entityToEdit);
            await axios({
                method: 'put',
                url: `${props.url}/${id}`,
                data: formData,
                headers: {'Content-Type': 'multipart/form-data'}
            });
        }
        else {
            await axios.put(`${props.url}/${id}`, entityToEdit);
        }
        history.push(props.indexURL);
```

Figure 13 - Axios Techonology 2

For replacing or improving an entire resource on a server, I implemented the Axios HTTP PUT method. (Figure 13 - Axios Techonology 2) Using an HTTP PUT request, which is frequently accompanied by an identifier or unique identifier that indicates the specific resource being changed, makes it possible for developers to communicate new data to the server. This method offers an efficient way to change or replace data on the server side and is regularly employed for full replacements of existing resources.

The POST and PUT methods are extremely significant for data submissions and resource modifications, respectively, even though Axios offers a variety of other HTTP methods like GET, DELETE, and PATCH. The aforementioned methods provide significant aptitude to control data activities in web applications through permitting effective communication between clients and servers.

4. The breakdown of the application

When this concept of designing and creating a web application designated for medical management showed up in my brain, I knew that it ought to be unquestionably patient-centered. That being so, as far as I am concerned, a use case diagram would be the most convenient way to provide details.

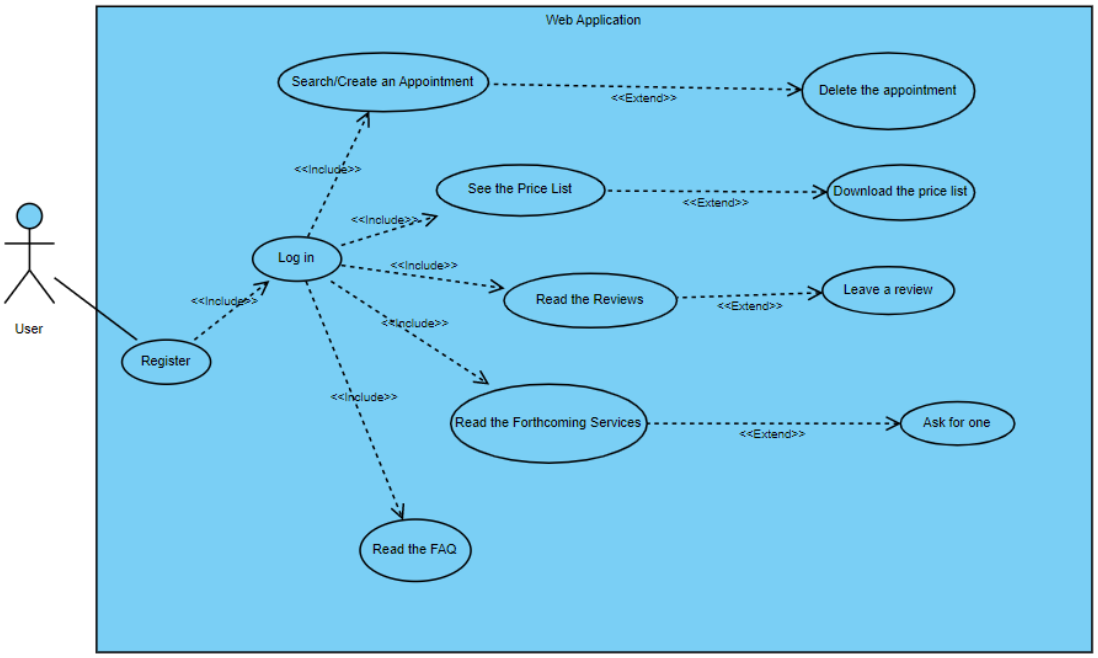

Figure 14 - General flow

This diagram Figure 14 - General flow cares for an overview of the application and what the user can do after the account is created.

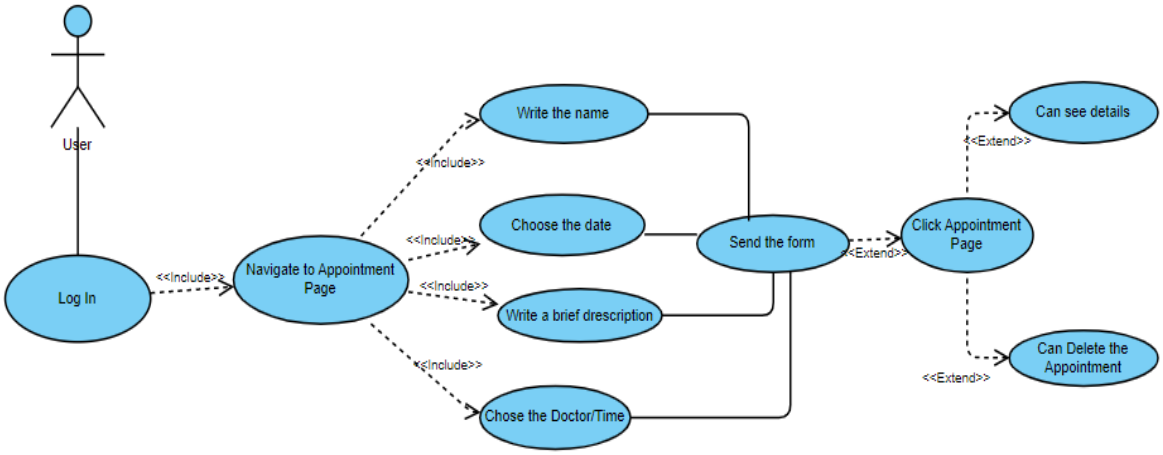

Figure 15 - Getting an appointment

The illustration Figure 15 - Getting an appointment sketches the process of creating an appointment and the details, the name, the wanted date, a brief description of the problem, and finally, yet importantly the doctor, details that are imperative for the form as a means for the appointment to be created and sent into the local database via the API. Nonetheless, if the user wants so, there is always the ability to see the created appointment and eventually delete it.

Description of the "Appointment" Case

Element of the Use Case	Description
Code	Every user will register with the email
State	Draft
Purpose	Create an appointment
Name	Appointment Process
Main Actor	User
Description	A user wants to appoint a medical consult with a doctor selected upon hers/his specialty
Preconditions	The user must have an account and complete the required fields in the form
Postconditions	N/A
Trigger	Click on Examination in the Menu
Basic Flows	The user ought to log in to the site using the credentials, being assumed that the account is already created. Then, the necessary fields displayed on the page need to be completed before the send button is submitted.
Alternate Flows	N/A

Relationships	<<Include>>/ <<Exclude>> Relationships
Frequency of use	It is being used for different situations: control needed/ emergency
Business Rules	The user wants to fill out the form

In another train of thought, based upon the fact that in the present circumstances, reviews matter when a choice is being made, from my personal standpoint, it seemed like a section dedicated to this would be rather appropriate. Nonetheless, the next figure will highlight this slice.

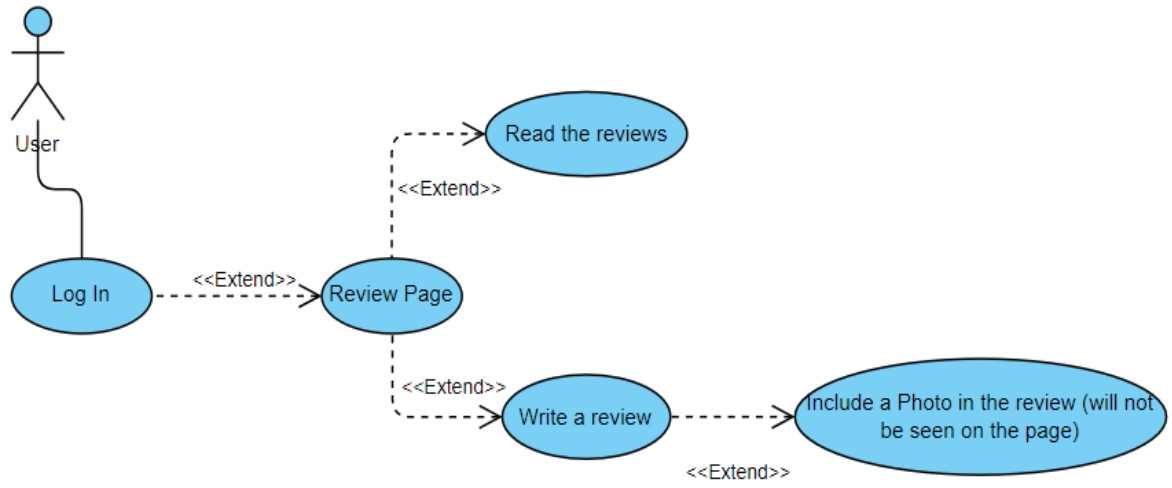

Figure 16 - Leaving a review

Narration Figure 16 - Leaving a review barely and revealing some information, this review section is to a degree rather distinctive, as well as a field for a photo to engage in. In a personal manner, I established that a photo with a doctor or of the clinic would be inclined toward being at least appealing.

Description of the "Review" Case

Element of the Use Case	Description
Code	Every user will register with the email
State	Draft
Purpose	Leave a Review on the site

Name	Leave a review
Main Actor	User
Description	A user wants to read the reviews about the clinic/doctors etc. There is always the option to leave a review
Preconditions	The user must be logged in to his account
Postconditions	N/A
Trigger	Click on Reviews in the Menu
Basic Flows	If it is wanted, the user can read the reviews from other people; He/she is also empowered at any time to write his/her beliefs and have the possibility to upload a photo with the doctor/clinic, etc. that will not appear in the Review part of the page
Alternate Flows	N/A
Relationships	<<Exclude>> Relationships
Frequency of use	It is being used with a singular purpose: read/rate the clinic
Business Rules	The user wants to participate in the overall rating

Offering a more comprehensive analysis, the very first paramount thing to mention is that my application does not deliver a great number of features when an individual is not registered or logged in to the site, as the majority of the site's capabilities are interconnected to one's personal account. This thinking pursues to some extent the concepts in this area of interest, as a virtual profile is essential (Regina Maria, MedLife…).

However, in order to complete the sign-up/log-in process, it is necessary to carefully go through the registration form and ensure that all security requirements are met, which cannot be avoided. There are two feasible outcomes that might occur at the time the send button is clicked: the first states that the registration form security requirements are met, and, evidently, the second scenario occurs when they are not fulfilled. Therefore, when registering, if the prerequisites are met, the piece of information containing the email and password is sent via the API into the local database. Otherwise, the registration form does not get submitted again unless it is modified.

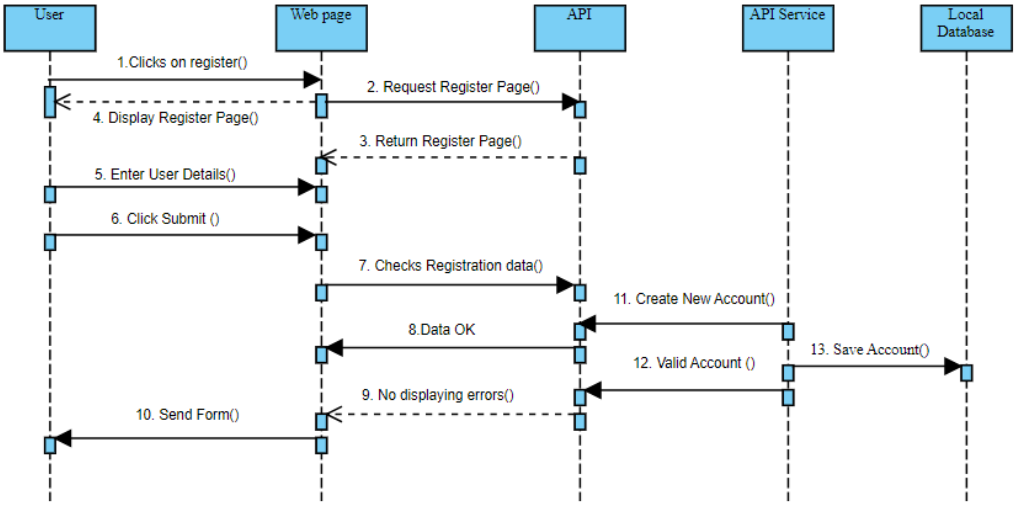

Figure 17 - Sign up case 1

As previously discussed regarding the two scenarios, the figure above Figure 17 - Sign up case 1 considers the first one and presents the stream of the procedure. Nevertheless, the user clicks on the register button from the menu, and while the duty is done in the background in nearly no consequential time, the page is loaded. Next, the form is filled out with the right information, and after the click, it gets sent to the local database in order for it to be stored. At this point, the user can now discover the site of their own choice.

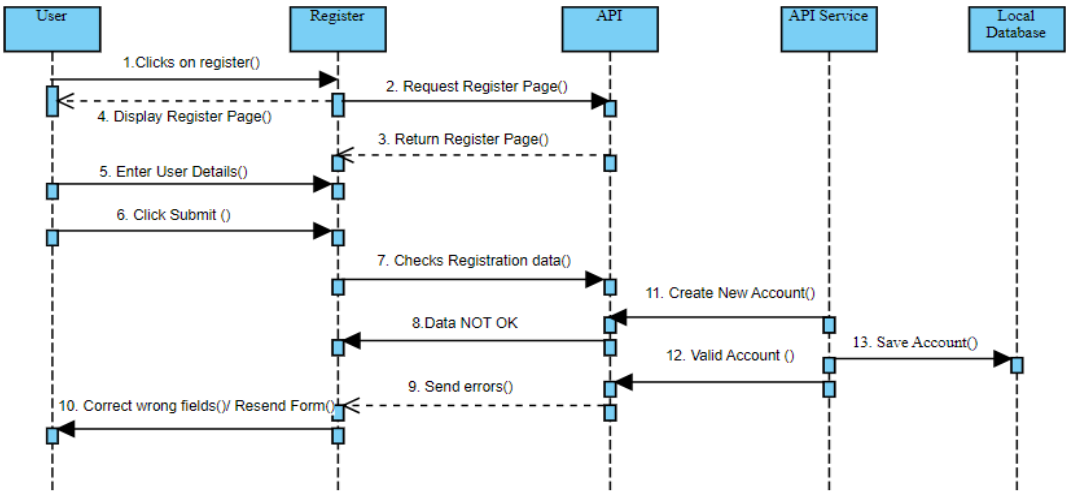

Figure 18 - Sign Up case 2

This succeeding scheme Figure 18 - Sign Up case 2 is responsible for the circumstance in which the user fails to measure up to the expectations for the account to registered. It is

discernible that there is a slight change in the flow of the events, in the end with the same aftermath.

Nevertheless, the local storage also has a salient word to say. Thus, I will introduce the diagram, and after that, I will explain more.

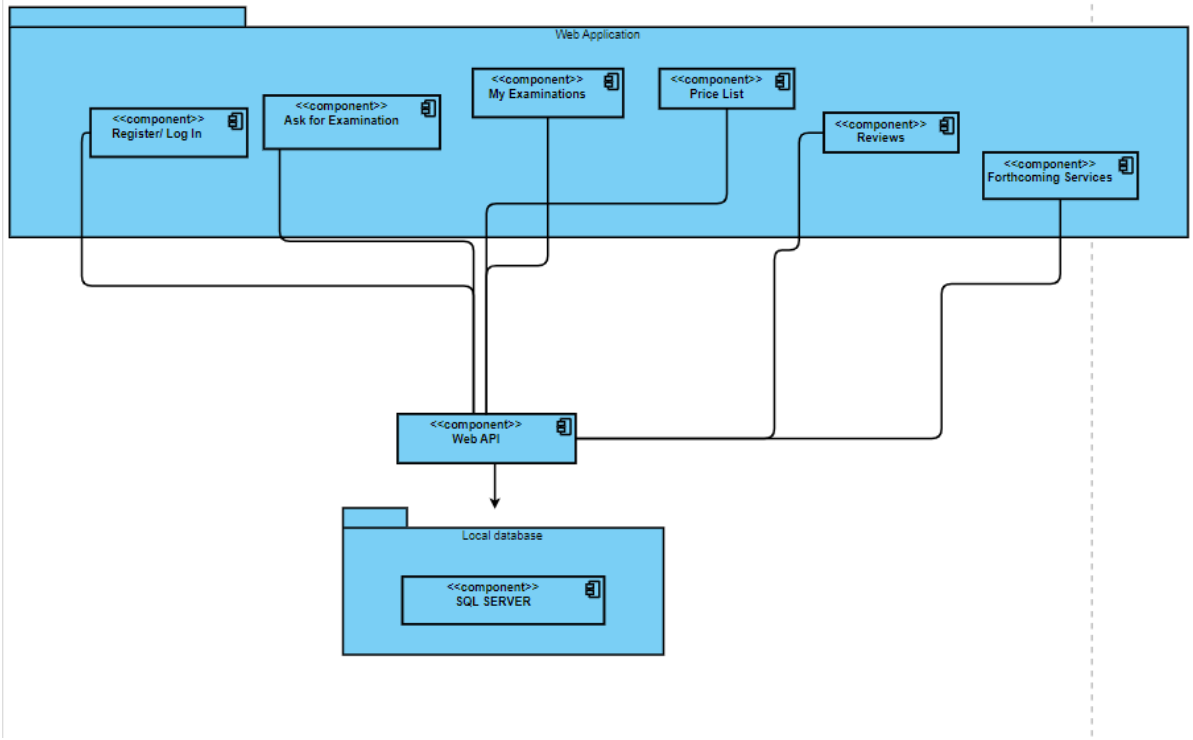

Figure 19 API - Database

To provide a more comprehensive analysis in Figure 19 API - Database, it is discernible that the application places a significant reliance on the database, being undoubtedly bound by it. Thus, the register and log in process, the form to ask for an appointment with a doctor, the list of the appointments, the price list, the reviews, and last but not least, the list with the forthcoming services that clients would like to see included in the clinic's list have an upfront way to transmit the insights to the database via the web API. Furthermore, as one would expect, this interconnection works in both orientations, as the database makes it possible for pieces of data like the price list to be shown on the application.

Come what may, when considering the multiple dimensions involved in the design of this app, it is crucial to underscore its central focus on addressing the specific needs and preferences of patients. An essential aspect worth emphasizing is the mandatory account creation and login process, which serves as a gateway for users to access the app's features. This prerequisite holds immense importance, as it not only ensures a personalized experience tailored to each individual but also guarantees a secure environment for utilizing the app's functionalities, fostering trust and confidentiality. By prioritizing patient-centered design and implementing robust authentication measures, the app strives to create a seamless and trusted platform for users to engage with and benefit from.

5. Modelling and describing the web application

When addressing the matter of discussing this certain nature of application, my modest point of view would say it would rather be uncomplicated, intuitive to use. I have the belief that at some point in time we all had an emergency, and inarguably you do not want to see a complex, somewhat harsh interface to use when there is suffering to bear; this is just a nimble example. On the grounds of this rationale stands the foundation of my silhouetting the interface of the app.

The earliest page Figure 20 - Register Page that ought to be accessed, emphasizing the fact that if the account is not created, ought to be the register page. Hardening the most crucial part of those accounted for above, I can state that the design is rather annoyingly elementary, with two fields that hold a label for what is expected to be written and a button.

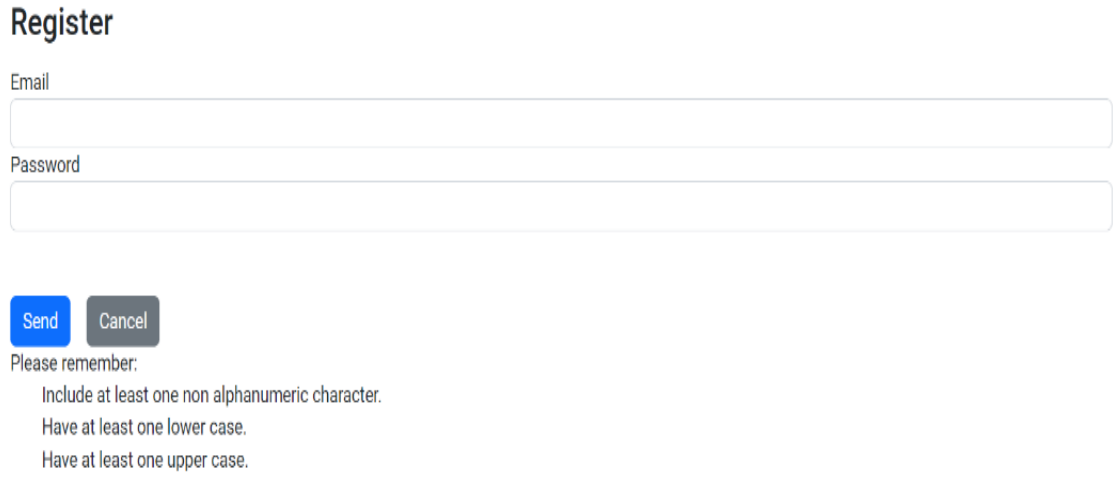

Figure 20 - Register Page

However, with the abundance of the twenty first century range of possible alternatives come twenty first century threats, and, I am referring to the cyber security ones. Keeping this in perspective, some non-negotiable requirements when referring to the password must be met for the account to be created, and those would be: a sizeable length, with at least one capital letter, and at least one special character. Furthermore, these restrictions come in a complementary manner with the email field, which ought to follow an email structure such as ...@yahoo.com. Moreover, in Figure 21 - Log in page the validation for the above-discussed

input is in real time: when the field is left empty, a red message “Required” will be displayed; As soon as a character is written into the email field, the message will metamorphose into “Invalid Email Address” as long as the format is incorrect. For this mechanism to have a closing point, the account will be stored into a local database, clearly with the password not in plain sight but hashed.

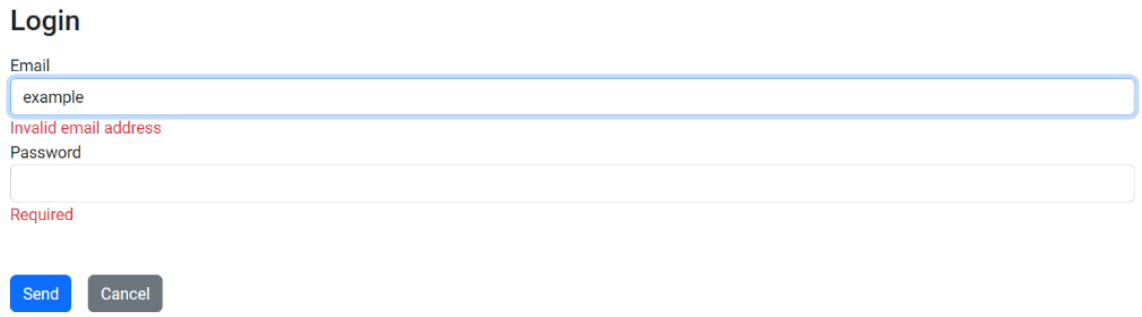

Figure 21 - Log in page

After this aftermath of this imperative mechanism of creating the account and logging in, if and only if the credentials are correct, the user will instinctively be redirected to the home page, and the site will disclose a handful of choices that are integrated in a menu on the top of the page, such as Ask for Examination, My Examinations, Price List, Reviews, Forthcoming Services and FAQ. Additionally, it’s worth mentioning that in the case of a logout miss click, the user will be shown a message that he/she is not allowed to see the page, foregrounding the significance of the account.

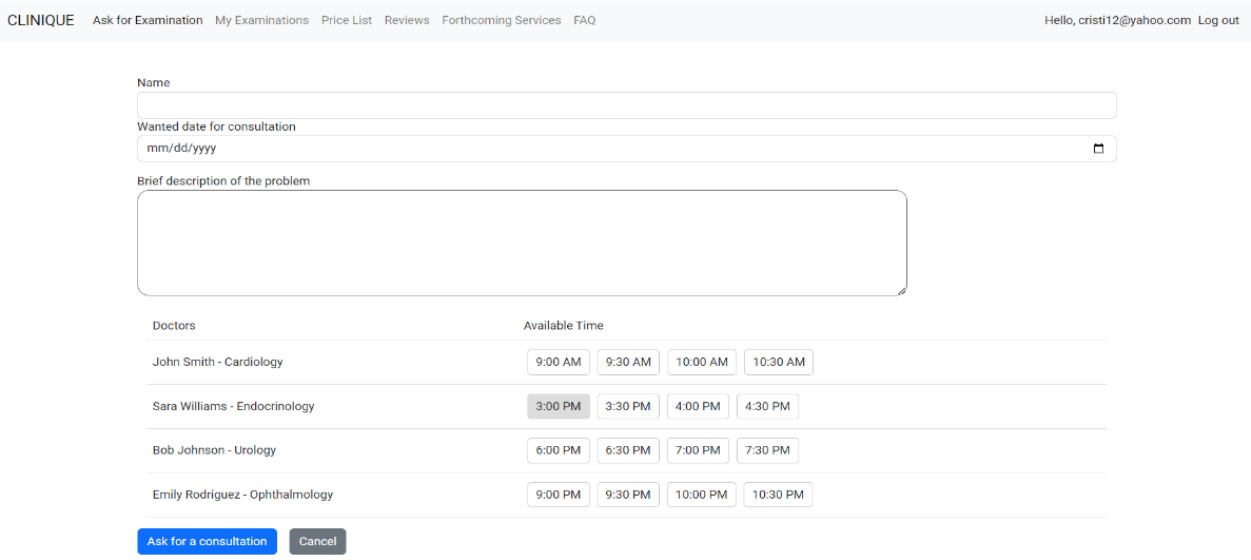

Figure 22 - Appointment page

The Ask for Examination section of the web app Figure 22 - Appointment page is, from my standpoint, one of the most salient components. As it can be observed from the picture, the page incorporates four categories in order for the appointment to be made: name, wanted date, a brief description that is not indispensable, and the list of doctors with their specialty and

schedule. In terms of validation, they were applied to the name field and the wanted date, as I consider them to be of the essence. Moreover, the fact that a past date can not be chosen is worth highlighting.

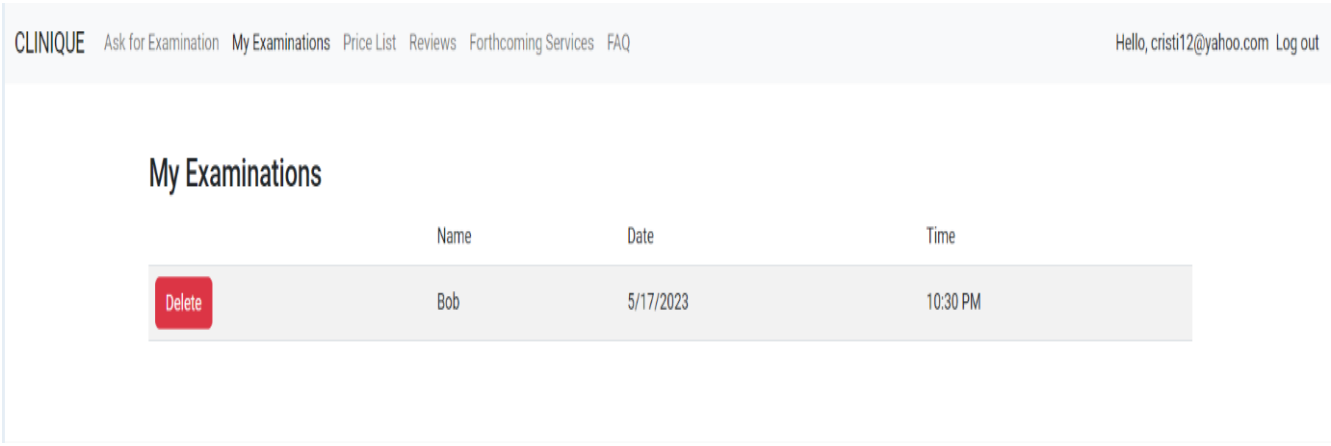

Figure 23 - My Examinations page

The following part Figure 23 - My Examinations page is, unmistakably, the one committed to the arranged meeting with the doctor. As can be seen in the figure above, a collection of, according to my way of thinking, the three most paramount pieces of information is shown. Also, in my humble opinion, the competence to delete the created appointment seemed to play a huge role in all this context.

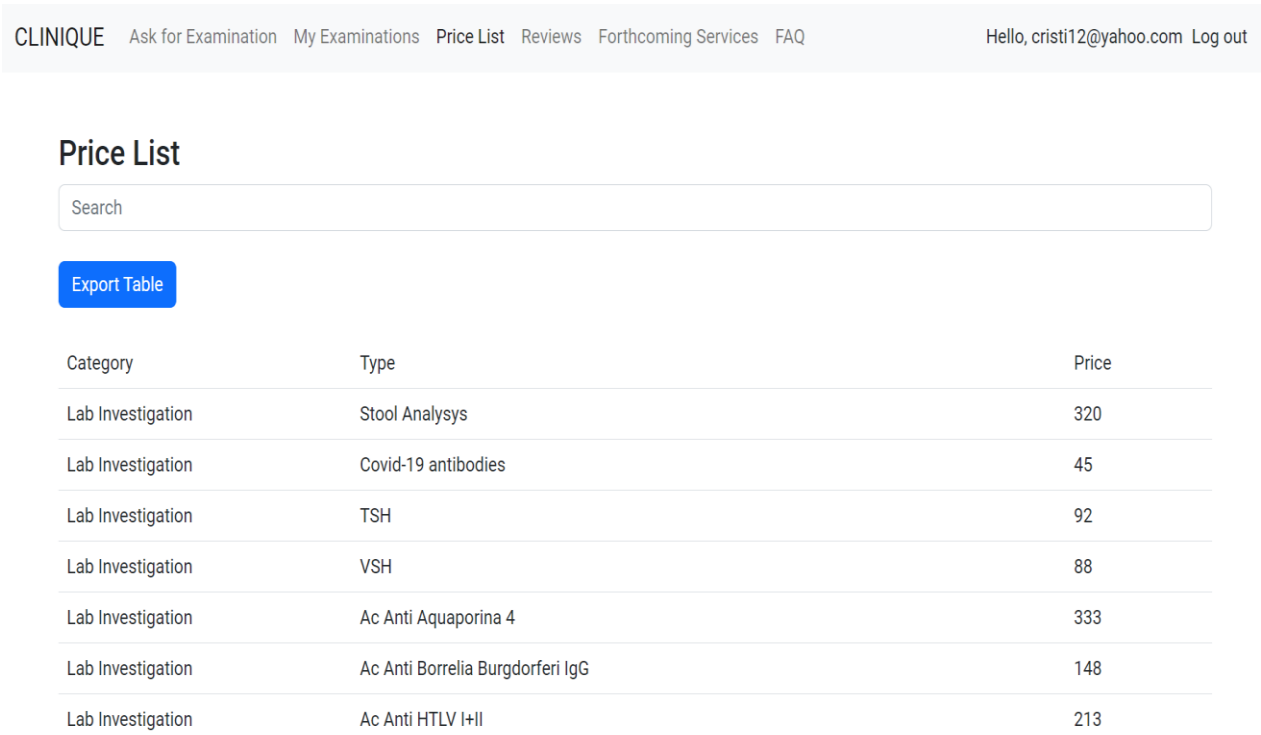

Figure 24 - Price List Page

There is also a page Figure 24 - Price List Page for the patient to check the price list. This may be very convenient for a wide range of individuals, as at this point in time more and more of us are more watchful with our budget. Furthermore, there is an option for the individual to export the whole table with prices to Excel format. In my humble opinion, this may be a very appreciable feature for those who need to analyze in detail.

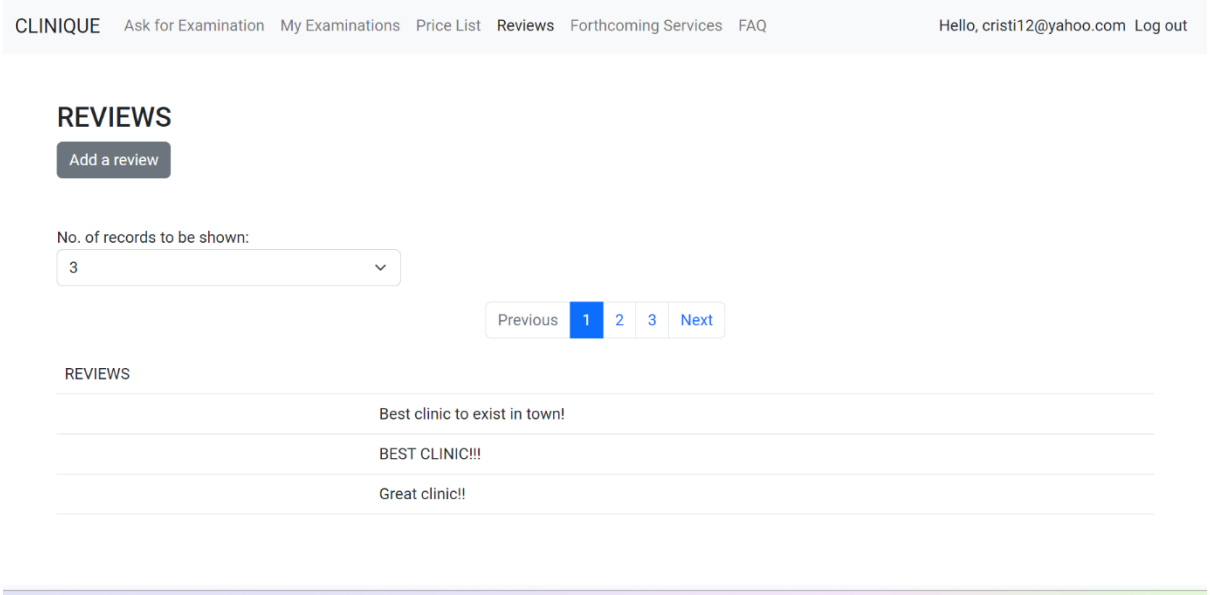

Figure 25 - Leave a review page

The reviews Figure 25 - Leave a review page section is pretty indicative, and it is needless to say that it has only one solitary yet salient role: to deliver the opinions of other patients. From my personal standpoint, I have the belief that this section should be accessed as it may inform others about the doctors, the services, and the clinic generally speaking. In addition to this, it is more indisputable that there is an option for one to also leave a review.

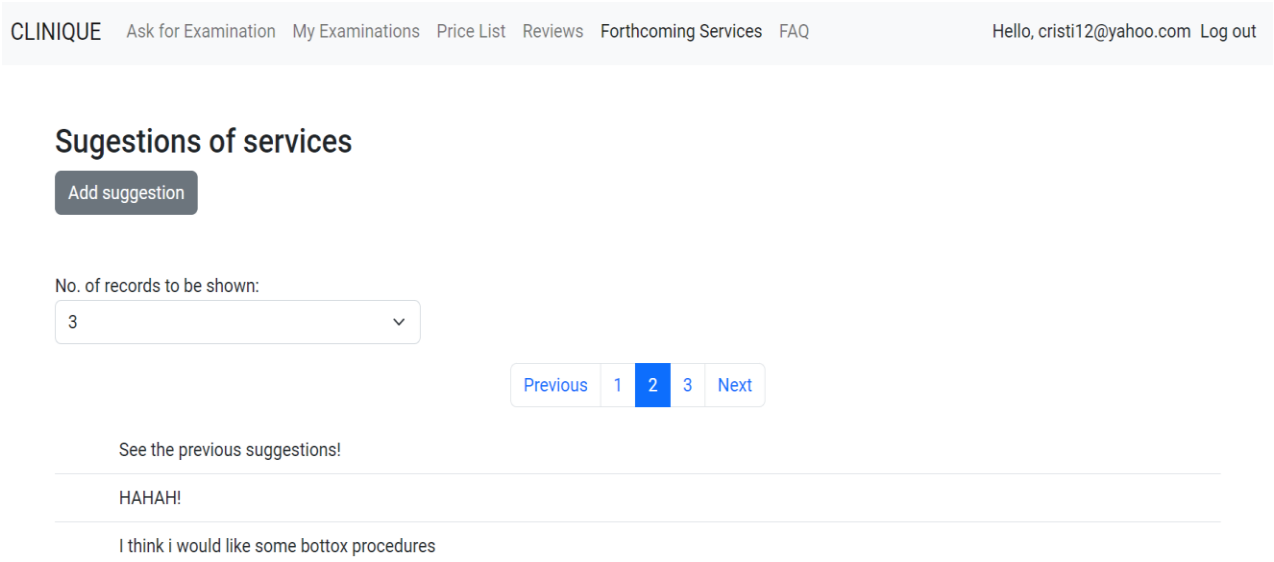

Figure 26 - Suggetion of future service page

Thinking of it as more of a personal touch Figure 26 - Suggetion of future service page , I had the belief that such an option, a list of forthcoming services that are desired by the patients, would be rather interesting and stimulating as it would imply a personal touch of people. Moreover, the roots of this idea lie in quite straightforward reasoning: one would like this clinic so much that he or she would like to come back for a particular type of service. This approach fosters a stronger patient-provider relationship by empowering individuals to express their specific needs and preferences, ultimately leading to a more tailored and satisfying healthcare experience. This approach fosters a stronger patient-provider relationship by empowering individuals to express their specific needs and preferences, ultimately leading to a more tailored and satisfying healthcare experience.

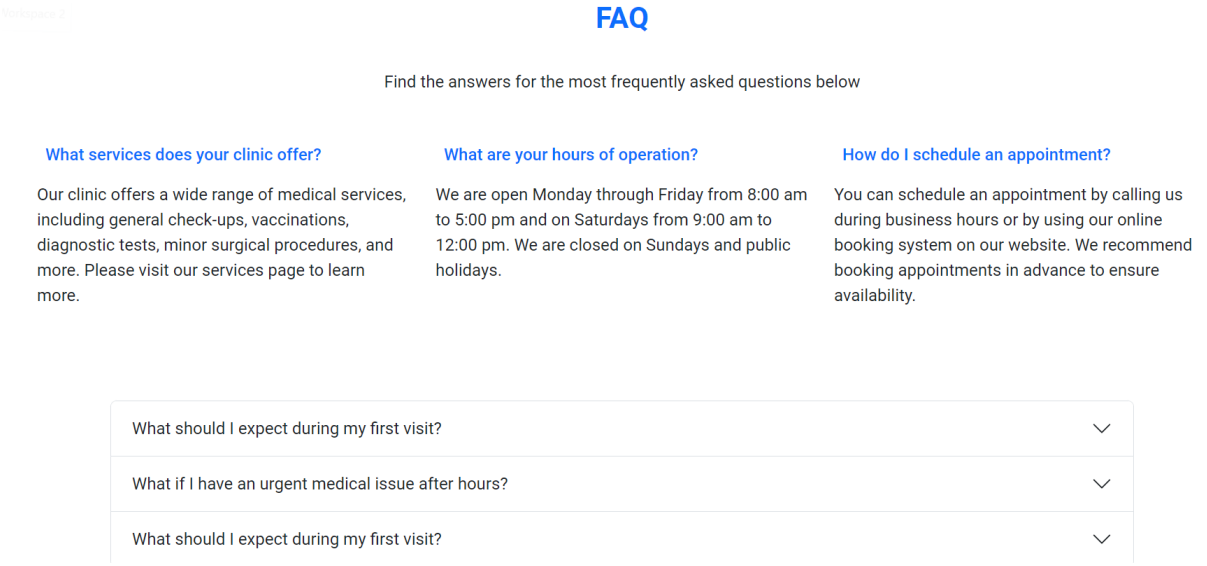

Figure 27 - FAQ page

Patients looking for information about a medical clinic and its services may gain insight through the Figure 27 - FAQ page 3 FAQ section of the clinic's website. It answers a lot of

questions about scheduling appointments, billing and insurance, clinic policies, treatments that are available, and various other issues. This area is of the utmost importance because it eliminates the need for patients to ask for essential information and allows them to quickly and easily locate solutions. It improves user experience, fosters trust, and makes sure that patients have access to the data they require to make safe healthcare decisions.

Conclusion

Reaching to the concluding part of this topic, from my personal standpoint, indeed it seems like through my application I managed to place a large amount of noteworthiness on the patient side, thus giving one the opportunity to take part in the future of the services provided by a medical clinic. I am convinced that this will result in a more enjoyable and effective experience for patients because it constitutes an important improvement in the way that medical clinics engage with their clients.

Nevertheless, as mentioned a salient number of times in this paper, cybersecurity also has a paramount word to say, one should also pay considerable attention to this, and I am speaking, of course, about a good, powerful password.

Another point that was made is the one regarding the digitalization of this whole process when an individual is seeking medical help in desperate times. On this account, waiting lists can be avoided. Furthermore**, digitalization can help improve the coordination of care.** By sharing patient data electronically, healthcare providers can better communicate with each other and ensure that patients receive the most appropriate care.

As a rewind:

- In the first chapter, I provided an analysis of the domain, looking it as a whole
- In the second chapter, I detailed the technologies I used to build this web application.
- In the third chapter, my attention was concentrated on how I modelled the app and I tried to care for the whole flow of the application.
- In the forth and last chapter, I presented the web application.

However, it goes without saying the fact that the application can suffer a lot of upgrades in order to add more functionalities.

Bibliography:

[1] “CSIRO PUBLISHING | Australian Health Review.” https://www.publish.csiro.au/AH/

[2] A. H. Seh *et al.*, “Healthcare Data Breaches: Insights and Implications,” *Healthcare*, vol. 8, no. 2, Art. no. 2, Jun. 2020, doi: 10.3390/healthcare8020133.

[3] “Global Misdiagnosis insides - medical error statistics by countries.” https://icloudhospital.com/articles/global-misdiagnosis-insides-medical-error-statistics-by-countries

[4] S. Health, “2021 Year-End Insights Report: $44B Raised Globally in Health Innovation, Doubling Year Over Year,” *Medium*, Jan. 10, 2022.

[5] “What is ASP.NET? | .NET.” https://dotnet.microsoft.com/en-us/learn/aspnet/what-is-aspnet

[5] “Set up Visual Studio Code.” https://code.visualstudio.com/learn/get-started/basics

[7] “Stack Overflow Developer Survey 2022,” *Stack Overflow*. https://survey.stackoverflow.co/2022/?utm_source=social-share&utm_medium=social&utm_campaign=dev-survey-2022

[8] “Introduction · Bootstrap.” https://getbootstrap.com/docs/4.1/getting-started/introduction/

[9] “Usage Statistics and Market Share of Bootstrap for Websites, June 2023.” https://w3techs.com/technologies/details/js-bootstrap

[10] “SQL Server Management Studio (SSMS) - SQL Server Management Studio (SSMS),” Mar. 31, 2023. https://learn.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms

[11] “What is SQL Server,” *SQL Server Tutorial*. https://www.sqlservertutorial.net/getting-started/what-is-sql-server/

[12] “Getting Started | Axios Docs.” https://axios-http.com/docs/intro