# Chapter 4: Implementation

## 4.1: Software Architecture

Our software architecture is a client-server architecture, and we implemented it as follows:

Our website is divided into two parts.

### The first part is the client

which we implemented with.NET 6 Razor Pages .NET 6 Razor Pages offers several compelling reasons for developers to choose it as their preferred framework. Here are a few key points highlighting why it is a good choice:

1. Simplicity and Productivity: Razor Pages embraces a minimalist approach, making it simple and straightforward to build web applications. With its convention-based model, developers can quickly create pages without the need for complex configurations. This simplicity enhances productivity, allowing developers to focus more on building the application's features rather than getting caught up in boilerplate code.
2. Seamless Integration with ASP.NET Core: Razor Pages is an integral part of ASP.NET Core, which provides a robust and versatile web development framework. It seamlessly integrates with other ASP.NET Core components, such as middleware, routing, and dependency injection, enabling developers to leverage the full power of the ecosystem while building their applications.
3. Powerful Templating Engine: Razor, the templating engine used in Razor Pages, offers a powerful and intuitive syntax that combines HTML markup with C# code seamlessly. This fusion allows developers to create dynamic web pages easily, with the ability to embed C# code directly into the views. Razor's syntax provides excellent readability, maintainability, and code reusability.
4. Flexible Page Model: Razor Pages follows the Page Model pattern, where each page has its associated code-behind file. This approach promotes a clear separation of concerns, making it easier to manage and test individual pages. It also supports the use of data annotations and model binding, simplifying the validation and processing of user input.
5. Lightweight and Performant: Razor Pages is known for its lightweight nature, resulting in fast startup times and improved performance. It has a small footprint, making it an excellent choice for resource-constrained environments or microservices. Furthermore, Razor Pages benefits from the performance optimizations introduced in .NET 6, enabling even greater efficiency and responsiveness.
6. Smooth Migration and Compatibility: If you're already familiar with ASP.NET Web Forms, migrating to Razor Pages is relatively straightforward. Razor Pages provides a similar programming model, allowing you to gradually migrate your existing applications while taking advantage of modern web development practices and technologies.
7. Vibrant Community and Ecosystem: Razor Pages is backed by a vibrant and active community of developers. It benefits from the extensive ecosystem of libraries, tools, and extensions available

for ASP.NET Core. The community support ensures that you can find resources, tutorials, and assistance whenever you encounter any challenges during development.

## The second part is the server

Here, our backend is divided into two projects

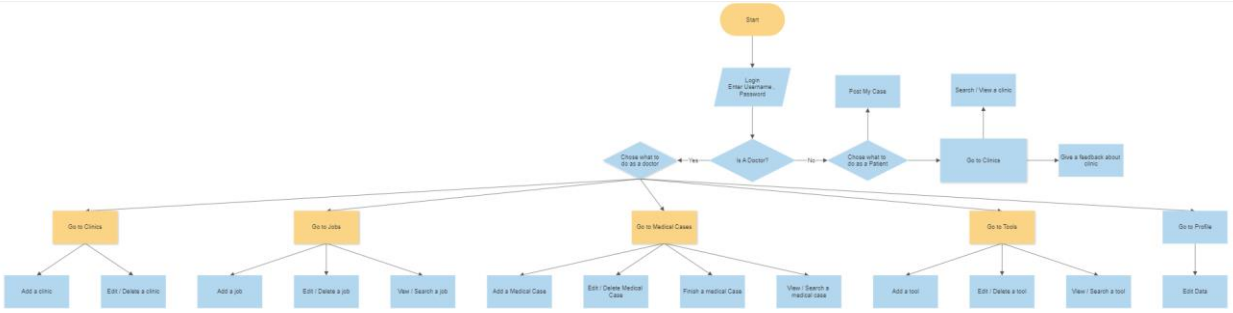
- .NET 6 Web API

NET 6 Web API provides a robust and versatile framework for building backend services, offering several compelling reasons to choose it for your application's backend:

1. High Performance and Scalability: .NET 6 Web API is built on top of the high-performance ASP.NET Core framework, optimized for handling heavy workloads and delivering exceptional performance. It leverages the latest enhancements in .NET 6, such as minimal API templates and performance optimizations, enabling you to build scalable and efficient backend services capable of handling large volumes of requests.
2. Cross-Platform Compatibility: .NET 6 Web API is a cross-platform framework, allowing you to develop and deploy your backend services on Windows, macOS, and Linux environments. This flexibility enables you to target a wide range of platforms, ensuring your API can be consumed by various clients, including web browsers, mobile applications, and IoT devices.
3. Rich Ecosystem and Tooling: .NET has a mature and extensive ecosystem with a wide range of libraries, packages, and tools. This ecosystem enables developers to accelerate development by leveraging existing components, such as authentication frameworks, database connectors, logging tools, and more. Additionally, Visual Studio and Visual Studio Code provide excellent integrated development environments (IDEs) with robust debugging and profiling capabilities.
4. Security and Authentication: .NET 6 Web API offers built-in support for authentication and authorization mechanisms, making it easier to secure your backend services. It supports various authentication schemes, including JWT (JSON Web Tokens), OAuth, and OpenID Connect. Additionally, the framework provides mechanisms for role-based access control (RBAC) and claims-based authorization, allowing you to implement fine-grained security policies.
5. Flexible Routing and Middleware: Web API in .NET 6 uses a powerful routing system that allows you to define flexible and customizable routes. This routing system enables you to map incoming requests to appropriate controller actions efficiently. Additionally, the middleware pipeline in ASP.NET Core gives you granular control over request processing, allowing you to add custom middleware for logging, error handling, caching, and more.
6. Strongly Typed and Productive Development: With .NET 6, Web API supports C# 10 language features, enabling developers to write clean, readable, and maintainable code. The use of strongly typed models and validation attributes enhances code correctness and reduces errors. Furthermore, features like automatic model binding, input validation, and content negotiation simplify the development process, leading to increased productivity.

- Flask API (Python)
  We used Flask API for a small part of our project only to handle our AI.

## 4.2: Flowchart



# Chapter 5: Testing

# Chapter 6: Results and Discussion

## 6.1 Results

- ### 6.1.1 Expected result

We wanted to make a platform that combined dental students, dentists, and patients all together.

We wanted to connect them with each other so all 3 can benefit from each other, and especially dental students, we wanted to make their lives easier as we know they got a lot of hard time in university so they can get patients to work on and build their CV, so our main idea was to make them able to find cases to work on our website, and that can happen in 2 ways: first, if the doctor finds a case but he doesn't need it, he will submit it on the website for another student, and second, by the patient, he can log in our website and make a post about his case and what he feels, and evern can take photos of themselves so other students can see and help them.

Also, students and doctors can buy, sell, and rent dental tools from each other on our website, and dentists can post job positions to find students to work for them and also post their clinic so patients can come to them.

Also, each student-dentist got their own profile, which can be used as a CV because any case they work on they can post in their profile with a description of what they did and before-and-after pictures.

For patients, like I said earlier, they can post their case.

but also find the nearest clinic for them and rate it with an AI that can read their feedback and give it a rating from -1 to 1.

-1 means that it's negative, and 1 means it's positive, and their feedback can be in Arabic or English, and the clinic will have an overall rating based on these ratings.

- ### 6.1.2 Actual results

We were able to achieve all that we wanted, and we made sure that only dentists and dental students could access cases with an excellent verification system that made sure of their identity and university card, and the website was fully working.

With AI, we made 2 AIs, one for English and the other for Arabic.

The Arabic one wasn't as efficient as the English one.

## 6.2 Discussion

The only difference was that with the Arabic AI, we couldn't make it as efficient as we wanted, and that's because there are no AI models that can work with Arabic very well to this day because of its complex nature.

# Chapter 7: Conclusion

We successfully made a website, or as we love to call it, a platform, where you can enter it in three roles, and each one of them can access different functions.

1. Dental Student
    - can find medical cases to work on with two different styles. The first is data entered by fellow students, so it's fully examined and ready to work on if it's useful for the student, and the other is data entered by a patient, and the student can contact the patient for more information or to set up a time to examine him.
    - can buy, sell, and rent tools from other students and dentists by viewing all tools on our website and contacting the buyer if he wants.
    - can find trainings and job positions advertised by dentists or fellow students and contact them if he is interested.
    - can build his CV on his profile by filling out his profile information and finishing medical cases from our website.
2. Dentist
    - can buy, sell, and rent tools from other students and dentists by viewing all tools on our website and contacting the buyer if he wants.
    - can post a training or job position that he controls.
    - can post his clinic with full information about it so patients can find him easily, contact him if they want, and also rate it.
3. Patient
    - can find the nearest clinic so he can go to it and rate it and see previous ratings.
    - can post his case with a description of what he feels and maybe a picture of his mouth so dentists can help him.

We can enhance our project by making it available for mobile applications too and making a better AI model for Arabic feedback.

# Chapter 8: Future work

- Make a better AI model for Arabic feedback.
- Make it available as a mobile application.
- Make a way of communication (chats) between seller and buyer in tools and dentists and patients.
- Make an automated verification system with an AI model that can verify students and dentists without the need for human interaction.