**Chapter 1: Project Presentation**

## Intoduction

In this chapter, we will start with a presentation of our project. Then we will determine the objectives to be achieved. Thus, we will focus on the study of the existing and its criticisms to propose appropriate and possible solutions.

## Project description

**What is web application?**

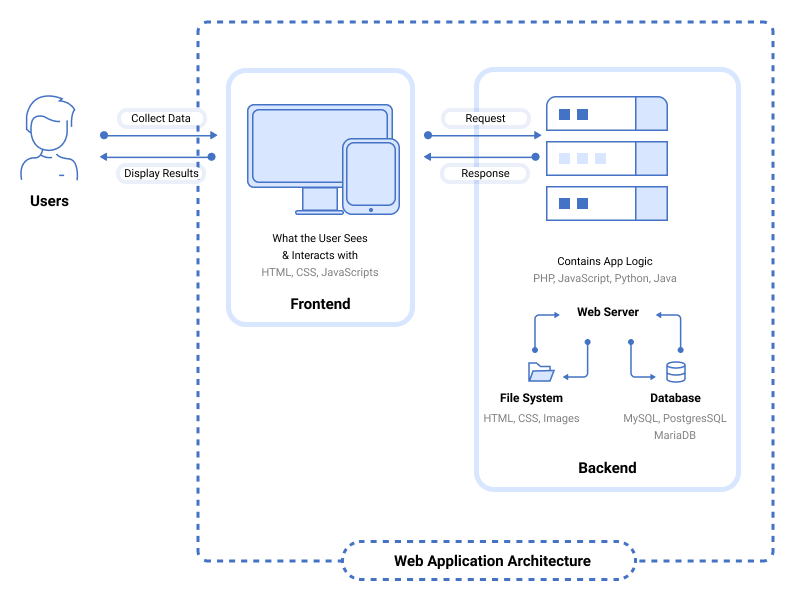
A web application is a software that runs in an Internet browser. Any website implying interactive elements can be called a web app. This means that the user can interact with the platform by pressing buttons, filling out forms, requesting a price or making purchases. Similar to desktop computer software or a mobile application, a web app provides a user interface, offers utility or entertainment, and the ability to access, create, store, or modify data.

**How does a web application work?**

All you need to access a web app is an internet connection. You use a web browser like Safari, Mozilla Firefox or Google Chrome to connect to your app. There are three elements the web application requires to function: a web server to handle requests from the client, an application server to execute the tasks requested and a database to store the information.

**Here is how a web application works:**

1. The user creates a request to the web server over the Internet through the application's user interface.
2. The web server sends this request to the web application server.
3. The web application server executes the requested task, then generates the results of the required data.
4. The web application server sends those results back to the web server (requested information or processed data).
5. The web server carries the requested information to the client (tablet, mobile device or desktop).
6. The requested information appears on the user's display.



## Project presentation

**What is a telemedicine**

Telemedicine is any use of electronic communications to get health care when you and the doctor aren’t in the same place at the same time. Telemedicine is actually a simple idea: Most people already have a smartphone or a personal computer, why not use it to speak with a doctor online? The benefits have been huge, giving people access to quick and convenient medical care for a wide range of health issues that don’t need to be seen in person.

**Problematic and critical of the existing**

From our research and observations, we concluded that most people find it difficult to access health services,especially people who live in the mountains and in remote areas. So, we decided to develop an application which allows to benefit from these services.

We have to develop this application because of:

* No online communication between doctor and patient.
* Facilitate the medical consultation to the person who had appointments with the doctor without having to travel for it.
* In fact, there is an already several applications worldwide that revolves around telemedicine but in our case, we noticed that in our country there is no telemedicine application.
* You don’t have to drive to the doctor’s office or clinic, park, walk or sit in a waiting room when you’re sick.
* Virtual visits can be easier to fit into your busy schedule.

**Goals:**

Indeed, our project consists of:

* Facilitating health services
* Telemedicine has also helped with a looming doctor shortage by making it easier for doctors to see more patients, more quickly reducing wait times.
* This facility can be used even during emergencies, which can become very handy at crisis moments.
* Through the app, you can message your doctor any time and from anywhere for constant monitoring of your health condition.

**Needs study**

* 1. Functional needs
* We will have two types of users: doctor and patient.
* Doctor and patient identification (Have an account).
* Patient choose his doctor from a window representing the list of doctors according to category.
* Patient must enter a password to access the doctor's area.
* Patient can ensure a reservation of an appointment with a doctor.
* Patient obliges to use online payment.
* Patient receives an automatic email after payment of the appointment.
* Patient receive a notification before the departure from meeting of 15 min
* Patient can import data (medical analyses).
* Application users can read the part that presents medical advice.
  1. Non functional needs
  + Be compatible with any operating system.
  + Developer telemedicine application desktop using Laravel.
  + Passwords are stored securely.
  + Privacy and security.
  + Performance.
  + Scalability.
  + Extensibility.
  + Availability.

## Conclusion:

In this chapter we presented the project in general, as well as the functionalities and the tasks that we will carry out, in the second chapter we will represent the phase of analysis and design of our project.

**Chapter 2: The application system design**

## Introduction:

This chapter outlines the conceptual solution that aims to make the task of management flexible and a projection of the physical reality. So we opted for the use of the UML notation which is a standard widely used in the analysis and design, this phase of design is carried out by several levels, in order to achieve a functional system reflecting a physical reality.

## UML Modeling:

UML, which stands for Unified Modeling Language, is a standard language for creating an integrated set of diagrams which help developers to specify, visualize, demonstrate and document the entire artefacts of the system to be developed.

**Class diagram:**

****

**Use case diagram**

****

****

****

**Sequence diagram:**

Create an account



Authentification



Search for a doctor



Make an Appointment



Consult the appointment



## Conclusion:

The conceptual phase is a fundamental step for the realization of any project. It allows us to facilitate the information system and carry out the implementation of the database and the processing. Then, we have to look for the possible means and tools to develop the application, which we will present in the next chapter.

**Chapitre 3 :** **Les outils de travail**

## Introduction:

The purpose of this chapter is to present the different tools and technologies implemented for the realization of the Application web. This technical study will show the richness and power of the technologies deployed to successfully complete the project.

## Languages:

**** **HTML5:**

HTML5 (Hyper Text Markup Language) is a system that allows the modification of the appearance of web pages, as well as making adjustments to their appearance. It also used to structure and present content for the web.

**CSS3:**

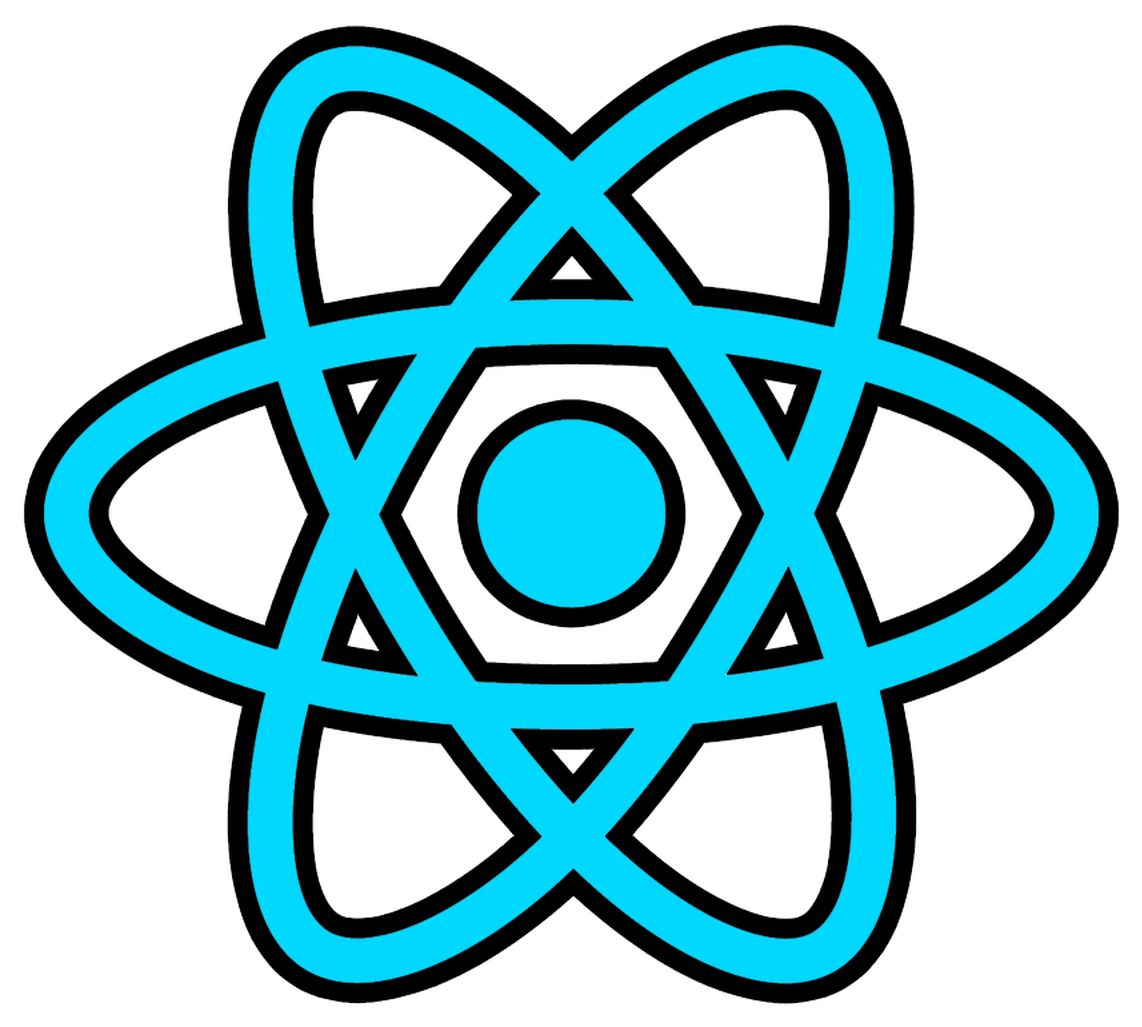
Cascading Style Sheets Level 3 (CSS3) is the iteration of the CSS standard used in the styling and formatting of Web pages. It is used with HTML to create and format content structure. It is responsible for colors, font properties, text alignments, background images, graphics, tables, etc. It provides the positioning of various elements with the values being fixed, absolute, and relative.



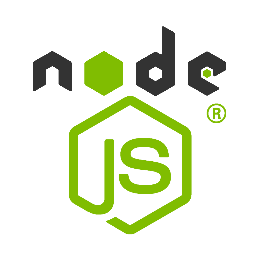
**JavaScript:**

JavaScript is a scripting language embedded in an HTML document. Historically, it is even the first scripting language for the web. It is a programming language that allows to improve the HTML programming language by allowing to execute commands on the client side, that is to say on the level of the browser and not the web server.

**Reactjs:**



Reactjs is an open-source JavaScript library that is used for building user interfaces specifically for single-page applications. It's used for handling the view layer for web and mobile apps.

**Nodejs:**

Nodejs (Node) is an open-source development platform for executing JavaScript code server-side. Node is useful for developing applications that require a persistent connection from the browser to the server and is often used for real-time applications

**Laravel:**



Laravel is a backend PHP-based and open-source Framework used for building a wide range of custom web applications. It's an entirely server-side Framework that manages data with the help of Model-View-Controller (MVC) design which breaks an application backend architecture into logical parts.

**Bootstrap:**

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains HTML, CSS and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

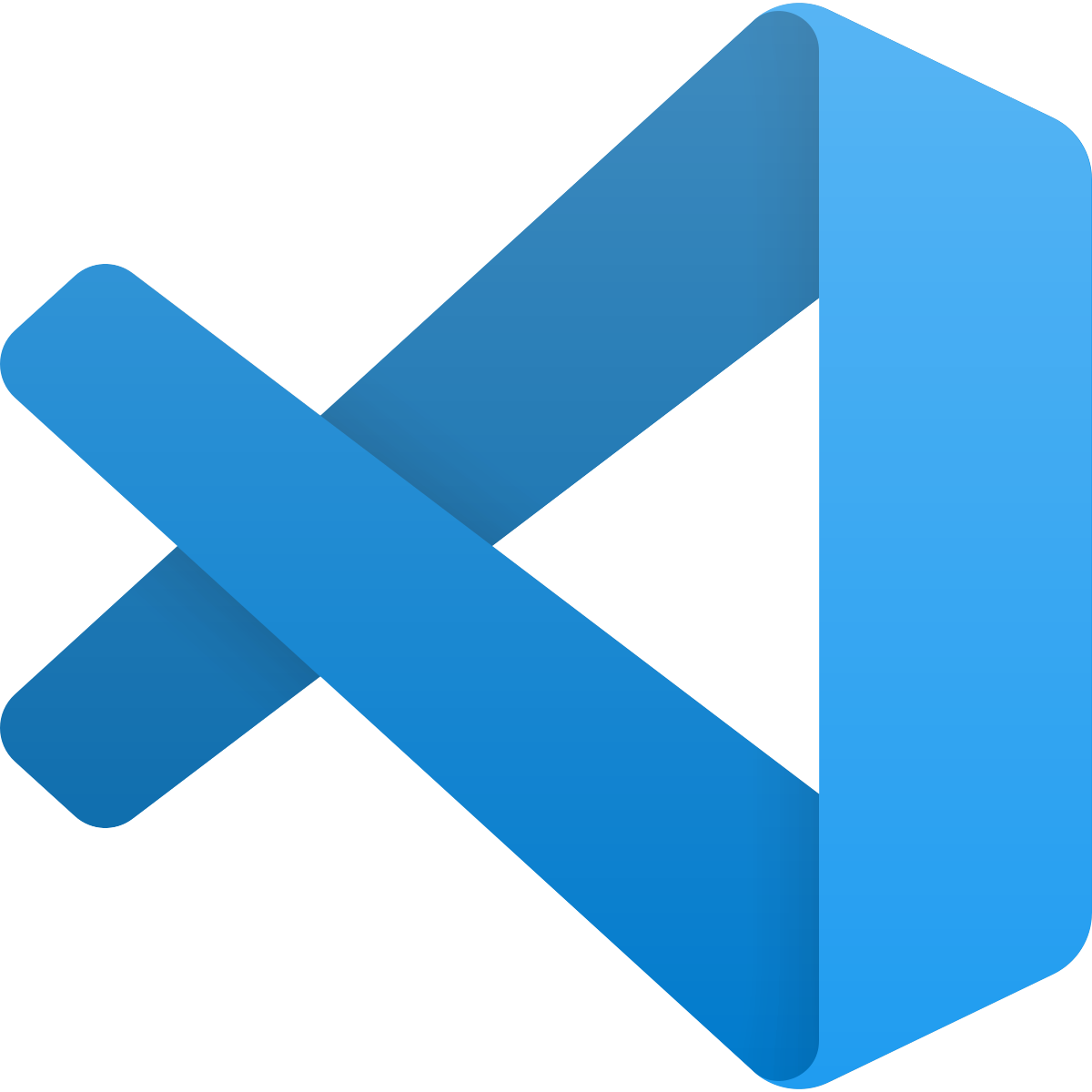
**MySQL:**

MySQL is a free relational database that was created in 1995 and is widely used on the Web, often in association with PHP (language) and Apache (web server). MySQL woks on all operating systems (Windows, Linux, Mac OS in particular). The principle of a relational database is to record information in tables, which represent groups of data by subject (customer table, supplier table, product table, for example). The tables are linked together by relationships.

**UML:**

UML (Unified Modeling Language) is a modeling language can be used to develop diagrams and provide users (programmers) with ready-to-use, expressive modeling examples.  
**Software:**

**Visual Studio Code:**



Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle

**Enterprise architect:**

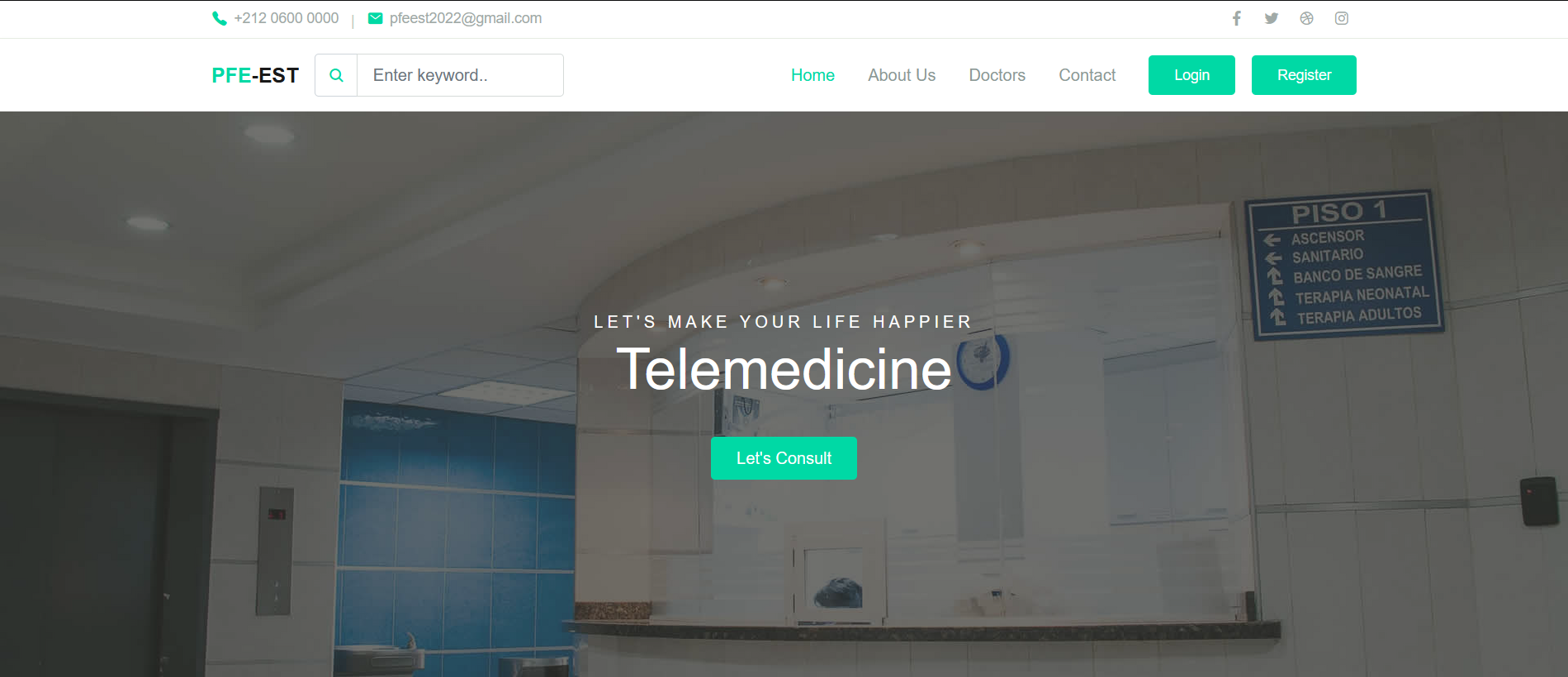


The most recognized UML modeling and design software. Covering by its functionalities, all the steps of the application design cycle.

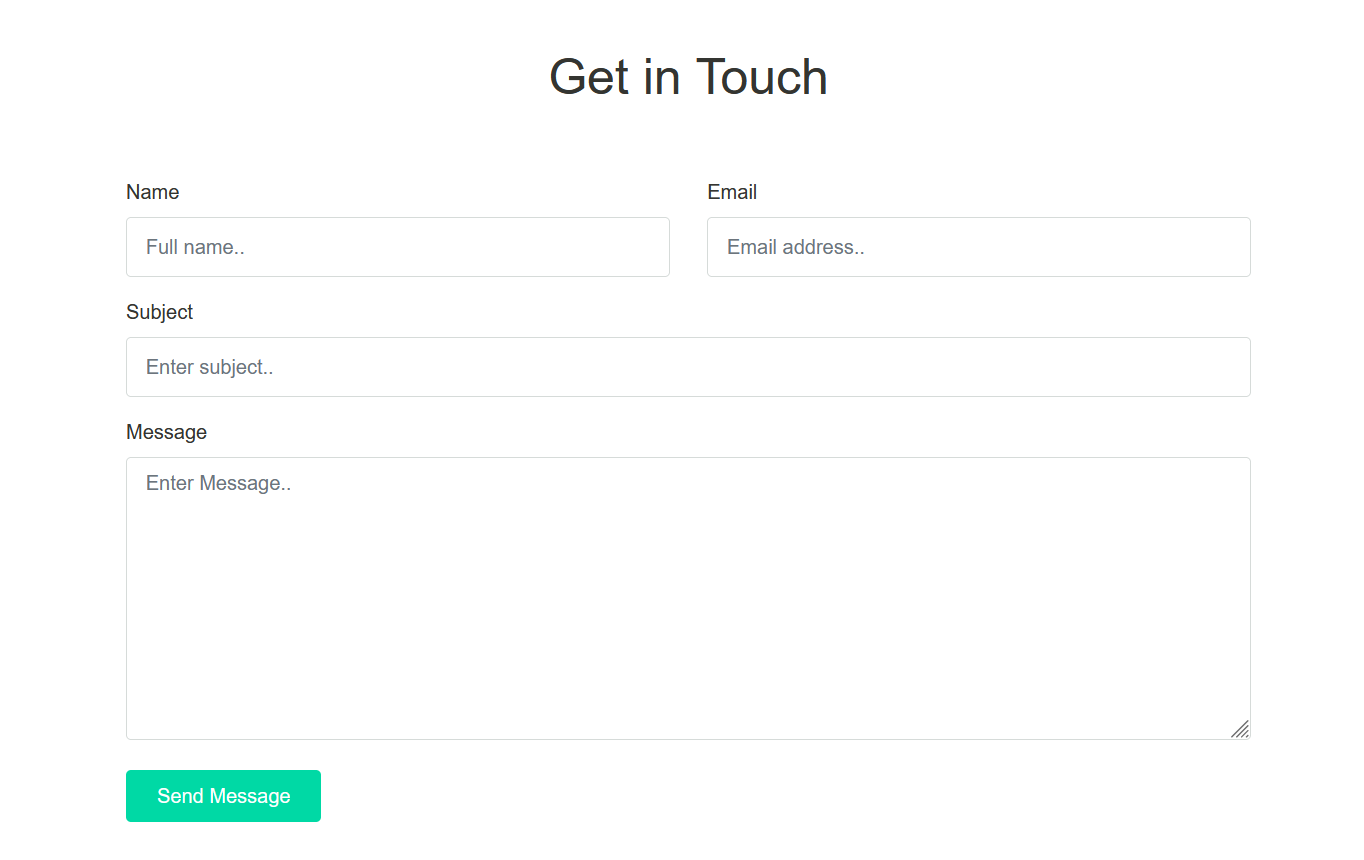
**Xampp:**



XAMPP is a software package that allows you to set up a local web server, an FTP server and an e-mail server. It is a free software distribution offering a good flexibility of use, known for its simple and fast installation**.**

****

**Contact us**

****