Kurdistan Region – Iraq
Ministry of Higher Education and Scientific Research
Lebanese French University
Department of Information Technology

Doctor Appointment System

A Graduation Project Submitted to the Department of Information Technology (IT) / Lebanese French University (LFU) as a Partial Fulfilment of the Requirement of the BSc. Degree in Information Technology

Prepared By:
Ari Subhi
Dara Jalal
Muhammed Samal

Project Supervised by:
Lec. Dr.Shakir Fattah Kak

2022 - 2023

Supervisor’s Certification

I certify that the preparation of this graduation research project titled “**Doctor Appointment System** “ was made under my supervision at the Department of Information Technology – Lebanese French University in partial fulfillment of the requirements for the degree of BSc. in Information Technology.

Signature:

Name:
e-Mail:
Affiliation: Department of Information Technology, College of Engineering and Computer Science, Lebanese French University, Erbil, Kurdistan, Iraq.
Date: / 05 / 2023

Examining Committee Report

We certify that we have read this graduation research project titled "**Doctor Appointment System** " and as an examining committee, examined the students:

In its content and in what is related to it, and that in our opinion it meets the standard of a graduation research project for the degree of BSc in Information Technology.

Examining Committee

Signature:

Name:
- Chairman -
Date: / 05 / 2023

Signature:

Name:
- Member -
Date: / 05 / 2023

Signature:

Name:
- Member -
Date: / 05 / 2023

Head of Department

Signature:

Name: Lect.Dr. Bnar Fareed
Date: / 05 / 2023

Dedication

This graduation research project is dedicated to: We dedicate this project to college department of computer engineering science and special thanks to Dr.Shakir Fattah , and we are especially grateful to our supervisor, and I also dedicate it to my department , who was able to complete our project successfully with their assistance.

1st Ari Subhi

2nd Dara Jalal Dara

3nd Muhammed Samal

Abstract

Designing a website for reserving data with a doctor is becoming increasingly important in today's world of digital healthcare. A good design for such a website should prioritize ease of use, security, and scalability to accommodate a large number of users. It should provide patients with a seamless experience that enables them to book appointments, communicate with their doctor, and access their medical records. Additionally, the website should comply with all relevant regulations and guidelines related to the storage and sharing of medical data, to ensure the confidentiality and privacy of patient information. A well-designed website for reserving data with a doctor has the potential to enhance the patient experience and improve access to medical care.

The goal of this project is to allow users to book appointments online through a mobile server or application, view the availability of doctors and their specialties, and receive reminders of upcoming appointments, making it easier for everyone. The first step in designing and implementing a website for reserving data and storing patient information is to prioritize user accessibility. Patients expect convenient and accessible healthcare services, and a website that offers a user-friendly interface can significantly improve their overall experience. The website should provide patients with the ability to schedule appointments, manage their medical records, and communicate with their doctors. The website should also be easy to navigate and should provide users with clear instructions on how to use its features.

Doctor Appointment System is an online platform that enables patients to schedule appointments with healthcare providers. This system allows patients to book appointments at their own convenience and helps them avoid long waiting times and the hassle of making phone calls. Through the doctor appointment system, patients can access their medical records, receive reminders for upcoming appointments and communicate with their healthcare providers. The implementation of a well-designed website for reserving data and storing patient information can revolutionize the delivery of healthcare services. The website can improve the efficiency of healthcare services, enhance patient satisfaction, and strengthen doctor-patient communication. Moreover, it can lead to more accurate diagnoses, better treatment plans, and improved patient outcomes.

The recommended system has been implemented and designed by utilizing various tools including PHP, CSS, HTML, XML, and Bootstrap for front-end web development, Xampp server, and the MySQL programming language for the database management system for the backend. These tools were chosen

for their versatility, functionality, and compatibility with the requirements of the system. In the end, the proposed website can improve the efficiency of healthcare services, enhance patient satisfaction, and strengthen doctor-patient communication. Moreover, it can lead to more accurate diagnoses, better treatment plans, and improved patient outcomes.

Table of Contents

Subject	Page No.
Supervisor’s Certification	I
Examining Committee Report	II
Dedication	III
Abstract	IV
List of Figures	VI
List of Tables	VII
List of Abbreviations	VIII
Chapter One: An Overview	
1.1. Introduction	1
1.2. The Aim	1
1.3. Related Work	2
1.4. Problem Statement	4
1.5. Proposed Solution	4
Chapter Two: The System Design	
2.1. Introduction	6
2.2. The System Architecture	6
2.3. The Data Flow Diagram	7
2.4. The Data Base	8
2.5. The Activity Diagram	10
Chapter Three: The Implemented System, Tests and Experimental Outcomes	
3.1. Introduction	12
3.2. Implementation Tools	12
3.3. Experimental Outcomes with Screen-shots	15
Chapter Four: Conclusions, Limitations and Future Works	
4.1. Conclusions	
4.2. Limitations	
4.3. Suggestions for Future Works	
References	
References	
Appendices	
Appendix : Backend	

List of Figures

Figure No.	Figure’s Title	Page No.
2.1	The Proposed System Architecture	
2.2	The Data Flow diagram	
2.3	The flow chart diagram	
3.1		
3.2		
3.3		
3.4		
3.5		
3.6		
3.7		
3.8		
3.9		
3.10		
3.11		
3.12		

List of Tables

Table No.	Table’s Title	Page No.
2.1		8
2.2		8
2.3		9
2.4		9

List of Abbreviations

List	**Abbreviations**
Data Flow Diagram	(DFD)
flow chart diagram	(FCD)
Visual Studio Code	(VS)
Hypertext Markup Language	(HTML)
Cascading Style Sheets	(CSS)
MY structure query language	(MYSQL)
Hypertext Preprocessor	(PHP)

Chapter One

Introduction

Chapter One

Introduction

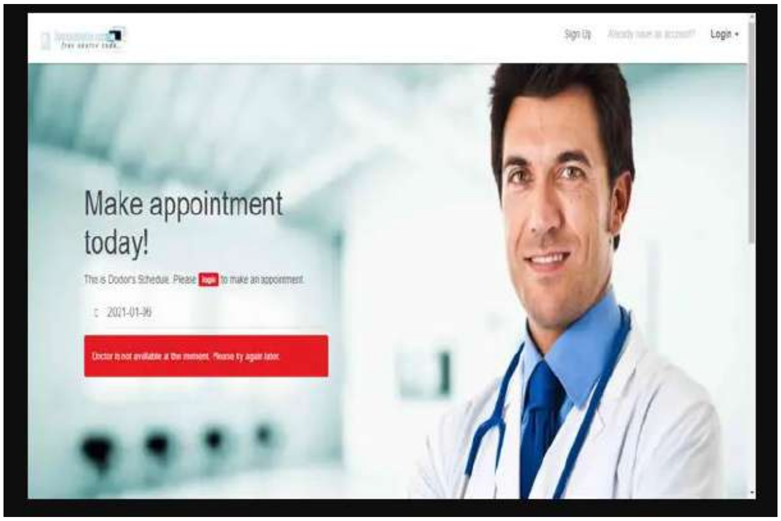
1. An Overview

Doctor reservation is an essential aspect of healthcare, and the online booking system has made the process more convenient and accessible for patients. Patients can easily book appointments online using their computers or smartphones, at any time of the day, without having to visit the hospital or clinic physically. This reduces the hassle of long waiting times, and patients can select the doctor and the timing that suits them the best [1][2].

The process of booking an appointment online is quite simple. Patients need to visit the healthcare provider's website, select the doctor they want to visit, and then choose the preferred date and time for the appointment. The online booking system provides real-time information about the doctor's availability, allowing patients to select the time slot that is convenient for them. After the appointment is booked, patients receive an automated email or SMS confirmation, which they can save and show at the hospital on the day of the appointment [3].

Moreover, online appointment booking also benefits healthcare providers, reducing administrative workloads and freeing up staff for more critical tasks. The digital booking system can also help hospitals and clinics manage their patient flow efficiently. By having real-time data about the number of appointments booked and the expected patient footfall, they can manage their resources effectively, ensuring that patients receive timely and high-quality care. Overall, online appointment booking is an efficient and convenient way for patients to schedule their doctor visits and improves the overall patient experience [1][4].

Online booking of doctor appointments has become a popular method for patients to schedule their medical appointments. According to a study by Software Advice in 2021, 48% of patients prefer online scheduling, and 77% of patients under 30 years of age prefer to book medical appointments online. This trend is expected to continue as more healthcare providers implement online booking systems [5]. Figure1.1 show an example of home page online booking system [5][6].



Make appointment today!

This is Doctor's Schedule. Please login to make an appointment.

2021-01-06

Doctor is not available at the moment. Please try again later.

Figure 1.1 Booking appointment doctor online main page.

The graph below shows the growth of online booking of doctor appointments from 2016 to 2020 in the United States. The data demonstrate a steady rise in the percentage of appointments booked online, with a significant increase in 2020 due to the COVID-19 pandemic [5] [6].

The benefits of online booking of doctor appointments are numerous. In addition to the convenience it offers patients, it can also reduce the workload on the medical staff, lower administrative costs, and increase patient satisfaction. Healthcare providers can also use online booking data to manage patient flow and allocate resources more efficiently [7][8]. Overall, the growth of online booking of doctor appointments represents a shift in patient behavior towards digital healthcare solutions. As more healthcare providers adopt online booking systems, the trend is expected to continue to rise, leading to a more efficient and streamlined healthcare system [6][7]. When a patient waits for long, the contagious diseases can be spread to the entire patient in the waiting room. Finding doctors in a particular field is a tedious task. The patient must consult the specified doctor for treating his/her disease. Figure1.2. Present the predicted total time for consult doctor [9].

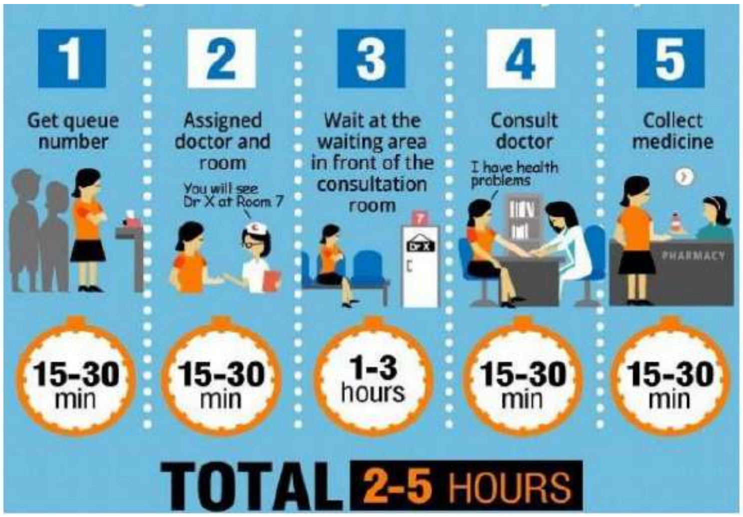


Figure1.2. Required Total Time for Consult Doctor.

By using this Web application, people can find a particular doctor in treating various kinds of diseases. It will make the patient stress-free and finish work on time and also reduces the spreading of diseases.

1.1. The Aim of the project

There are several objectives of the current project as listed below:

1. User-Friendly Interface: The website should have a user-friendly interface that is easy to navigate and provides a seamless booking experience for patients.
2. Mobile Compatibility: The website should be mobile-responsive, meaning it should be easily accessible and usable on all mobile devices. As more people use their smartphones and tablets to access the internet, a mobile-friendly website is critical to ensure patients can book appointments from anywhere and at any time.

3. Customizable Booking Options: The website should offer customizable booking options, such as the ability to select a preferred doctor or appointment time. This helps to ensure that patients can book appointments that are most convenient for them and improves overall satisfaction.

By considering these aims when designing a doctor appointment booking website, healthcare providers can provide a convenient and efficient booking experience for patients, while also improving the efficiency of clinic operations.

1.2. Related Work

The literature survey has discussed various applications of online booking doctor's systems. Also, a collection of recent works related to the project topic and the comparison process are presented. The objective of this thesis is to maximize the number of patients seen each day and a number of patients assigned to their top preferences. Scheduling urgent patients is also a consideration in this thesis. Some related papers are reviewed as follows. The published research [10] by Wang and Gupta considered patients' preferences and acceptable combinations of physicians and time blocks. They estimated patients' preferences in terms of acceptance probabilities, which contained different combinations of date, time, and physicians. Second, they assumed an online appointment scheduling system is applied in which patients selected the one preferred date, time blocks, and physicians. After receiving the request, the clinic scheduled one combination of date, time and physicians. If the clinic responded that none of the combinations are available, patients can repeat the booking process until they were assigned to an available combination. Two approaches (policies) were presented associated with decision-making.

The research presented by Vermealen et al. [11] studied an online appointment system considering the different urgency of patients and their preferences. This paper considered the situation when a patient made an appointment diagnosticians test. The objective was to assign patients before their next consult date with the physician. Non-urgent patients were assigned based on minimum access time and urgent patients were assigned to any timeslots left over on days before minimum access time. When considering patients' preferences, three boolean-type preference models were considered work/non-work hours on one day, multiple preferred days, and a combination of the previous two.

At present, there is a scarcity of studies addressing the pre-diction of no-show-up appointments. Most papers describe the use of one parametric model, for instance, the use of ordinary least squares to predict on a given day how many no-show-up appointments will occur. Logistic regression for binary classification is used to predict appointment misses of the patients. Most studies use very few features and apply limited analysis. Few studies developed regression models to predict appointment non-adherence. Some retrospective studies also worked on predicting no-show-up appointments. But that applies to a small dataset of a few thousand records.

1.3. Statement of the Problem

The traditional system of booking doctor appointments has long been associated with long wait times, missed appointments, and poor patient experiences. Patients are often required to make phone calls during business hours to schedule appointments, which can be time-consuming and frustrating. In addition, the system can be inefficient and costly for healthcare providers, with staff spending significant time on administrative tasks related to appointment scheduling.

Therefore, there is a growing need for modern, digital solutions that enable patients to easily book and manage their appointments online, while also providing healthcare providers with a more efficient and cost-effective way to manage their operations. This requires an understanding of the challenges and limitations of existing appointment booking systems, as well as the development of new technologies and strategies that can improve the overall patient experience and ensure the continuity of care.

1.4. Proposed Solution

The objective of the suggested system is to enhance the current facilities by addressing their shortcomings. The suggested system has the potential to overcome all the limitations of the current system by offering adequate security and minimizing the need for manual work. The current system has numerous drawbacks and limitations, which makes it challenging to function effectively. The suggested system seeks to eliminate or at least reduce these issues to some degree. As a result, the suggested system can alleviate the workload and mental stress of users. It enables users to work more efficiently in a user-friendly environment, allowing them to carry out their tasks without experiencing delays.

1.5 Layout of the Thesis

The project has the following layout:

Chapter Two Explains the principal concepts of designing and booking doctor appointment systems, web technologies, and a discussion about websites and databases.

Chapter Three Presents the proposed doctor appointment system services and website interfaces. Then, it explains the steps of implementing these techniques in building the website schema.

Chapter Four Presents the evaluation and assessment of the proposed system and its end-user interfaces.

Chapter Five Presents the conclusions for the proposed system and suggestions for future works.

Chapter Two

Background Theory

2.1 Introduction

Online booking appointments in healthcare is a growing trend that offers numerous benefits to patients and healthcare providers alike. With the increasing use of technology in healthcare, many practices and medical facilities have implemented online booking systems to allow patients to schedule appointments quickly and easily.

One of the main advantages of online booking appointments in healthcare is convenience. Patients can book appointments from anywhere at any time, without the need for phone calls or in-person visits. This can be particularly beneficial for patients with busy schedules, mobility issues, or those who live far away from the medical facility.

Online booking systems also offer improved efficiency for healthcare providers. With automated appointment scheduling and reminders, medical staff can spend less time on administrative tasks and more time providing patient care. Additionally, online booking systems can help reduce the number of missed appointments, which can save time and resources for medical facilities [12].

Another benefit of online booking appointment in healthcare is increased patient engagement. Patients are more likely to take an active role in their healthcare when they have the ability to book appointments easily and access their medical information online. This can lead to better health outcomes and improved patient satisfaction.

Overall, online booking appointments in healthcare is a convenient, efficient, and effective solution for patients and healthcare providers. As the healthcare industry continues to embrace technology, online booking systems are likely to become more prevalent and offer even more benefits to patients and providers [13].

2.2 The objectives of online booking in healthcare centers

The objectives of online booking appointments for doctors' centers are to improve patient access to healthcare services, streamline the appointment scheduling process, increase operational efficiency, and enhance patient satisfaction. Below are some specific objectives that online booking appointments for doctors' centers can help achieve:

1. Improve patient access to healthcare services: By offering online booking appointments, doctors' centers can make it easier for patients to schedule appointments at their convenience. Patients can book appointments outside of regular office hours, reducing the need for phone calls or in-person visits.

2. Streamline the appointment scheduling process: Online booking systems can help doctors' centers manage appointments more efficiently, reducing wait times and administrative workload. By automating appointment scheduling and reminders, doctors' centers can free up staff time for more critical tasks.

3. Increase operational efficiency: Online booking appointments can help doctors' centers optimize their resources and reduce costs associated with no-shows and missed appointments. By providing real-time availability, doctors' centers can maximize their appointment schedules and reduce gaps in their daily operations.

4. Enhance patient satisfaction: Offering online booking appointments can improve patient satisfaction by providing a convenient and efficient way for patients to schedule appointments. Patients appreciate the flexibility and convenience of online booking, which can contribute to a more positive overall experience.

5. Improve patient engagement: By providing online access to appointment scheduling and medical information, doctors' centers can

engage patients in their healthcare and empower them to take a more active role in their health management.

In summary, the objectives of online booking appointments for doctors' centers are to improve patient access, streamline operations, reduce administrative workload, enhance patient satisfaction, and improve patient engagement in healthcare management.

2.3 Waiting Time and Patients’ Appointment System

Waiting time simply means a period of time which one must wait in order for a specific action to occur, after that action is requested or mandated. Patients’ waiting time has been defined as “the length of time from when the patient entered the outpatient clinic to the time the patient actually received his or her prescription”. A patient appointment system or appointment schedule for health care center started long time ago. Management of patients’ appointments has earlier worked and has developed simplified queuing models and fairly static scheduling conditions.

2.4Appointment Delay

The appointment delay which is defined as the time between the day a patient requests an appointment and her actual appointment date, the higher the chances that he/she will cancel or not show up. This suggests an obvious way of minimizing no-shows and cancellations: this is done by asking the patients to come right away or make appointment requests on the day they want to be seen.

2.5 Managing Patients’ Appointment system

Managing patient appointment system is a computer application used to manage and reduce the patient waiting time in the health care center. Some health care centers do not use any appointment system. So, it has a longer average patients’ waiting time than the health care center that adopts the patients’ appointment system. While patients can wait for more than one hour to be attended to by a physician in a health care center, they also can feel that they are being disregarded and treated unfairly

2.6 Dynamic Web Content

Web technologies have been increasingly important as a way to promote the fast-gaining impact of the Internet as a marketplace for e-commerce. The composition of various controls used for the display and the collection of the information is a web-based interface.

Web applications are known as applications running on the internet; these applications have been designed to provide information and services via various technologies likes JavaScript, HTML, AJAX, JSP, PHP.

The web platform consists of a diverse architecture of several components and technologies including HTTP, website-server, server-side application development technology (Computer-Generated Imagery (CGI), PHP, and Active Server Pages (ASP)) and client-side technologies (such as JavaScript).

Furthermore, Web applications should be able to provide data security to overcome threats. Therefore, it should provide the specified functionality to prevent attacks utilizing different protection techniques as follow:

1. Authentication and approving: Authentication and authorization by using Kerberos, granular access control laws, etc., should be given to web applications.

3. Instant Secure Sockets Layer (SSL): Web applications should transform unsecured HTTP web application request automatically into encrypted HTTPS method.

2.7 Comprehension of unique features of web applications

The web application is a client-server application running through a web platform. Nowadays, web environment systems allow service delivery and dynamic information to users. The web application on the server accepts user input from its client (i.e. browser) via HTTP request, communicating with local file systems, backend repositories, and other data access; and data recall elements (Figure 2.1). However, HTTP responses send outputs (i.e. HTML pages) to the client .

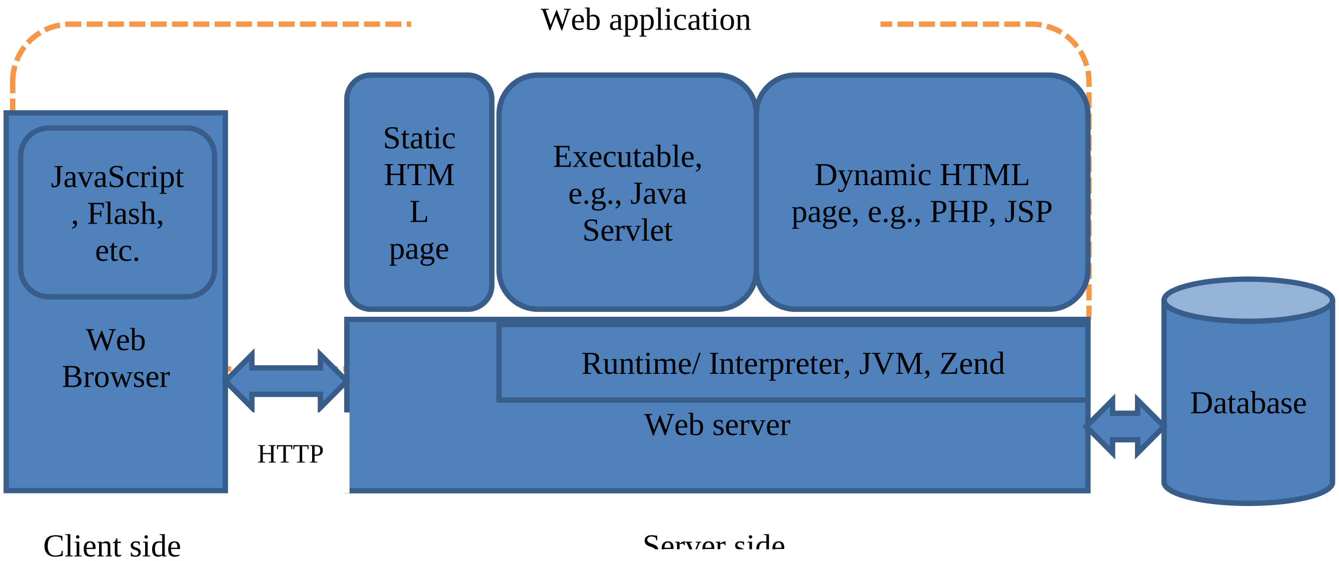


Figure 2.1: Web application description.

Chapter Three

The System Design

3. The System Analysis Design

3.1Introduction

The conceptual design of the proposed system was carried out using the data flow diagram , System Architecture , flow chart diagram as shown in the following parts.

3.2The System Architecture

The system architecture is the foundation of the solution and should be presented first. The core components will be indicated, piecing it all together in the overall architecture, with some thoughts about communications, showing the technical integration of the components and added features.in the system architecture of our project represent the general structure on which the web site is designed , it consists of two parts :

- The administration part (backend) : the persons authorized to access this part are the admins who work on the system and they enter .
- User part (frontend) : this part is intended for the user of the web site who searches for a products items .

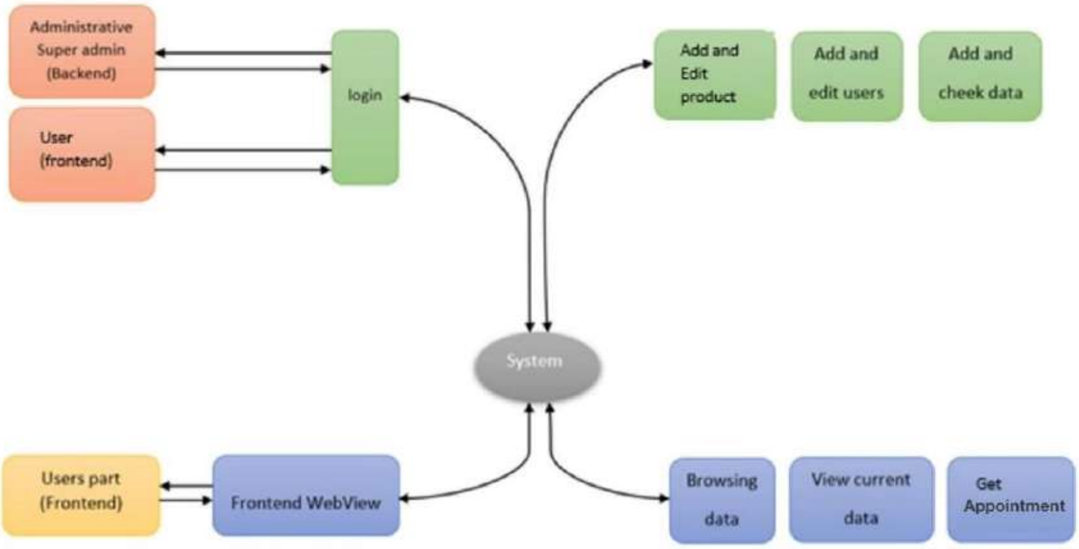


Figure 2.1 : The Proposed System Architecture.

3.3The Class Diagram

A class diagram is a type of static structure diagram in the Unified Modeling Language (UML) that describes the structure of a system by showing the system's classes, their attributes, methods, and the relationships among objects. It is used to model the object-oriented programming concepts such as inheritance, abstraction, polymorphism, and encapsulation.

In a class diagram, each class is represented as a rectangle, which contains the class name, its attributes, and its methods. The attributes are the data members of the class, while the methods are the operations that can be performed on the data. The relationships among the classes are shown using arrows between the rectangles. The most common types of relationships are inheritance, association, aggregation, and composition.

Class diagrams are useful for visualizing the structure of a system and can be used as a blueprint for designing or implementing a system. They are also helpful in communicating the design of a system to stakeholders, including developers, architects, and project managers.

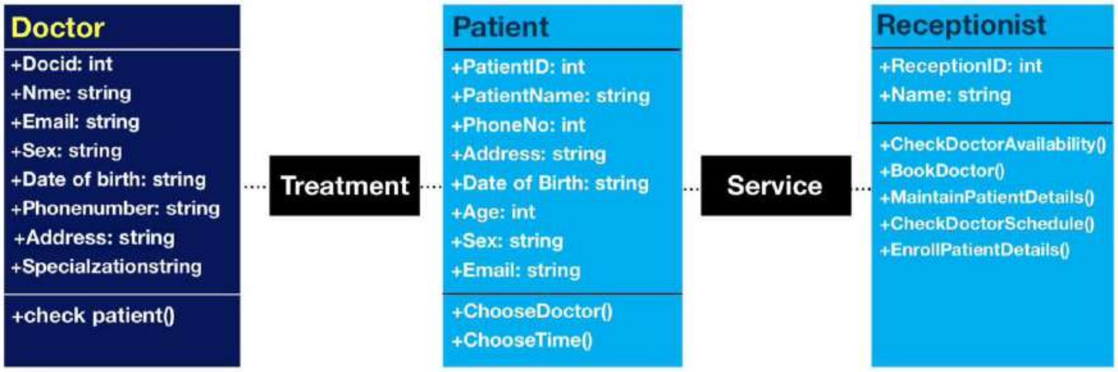
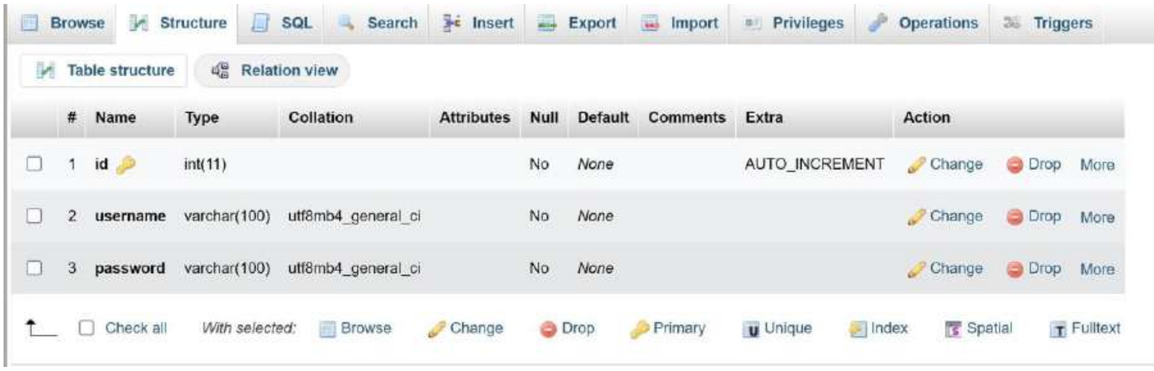


Figure 2.2 : The Class Diagram.

3.4Database Tables

Various associated tools, including MYSQL, were used for the implementation of the project's practical aspect. The XAMPP cross-platform was used to design the system database tables, which are depicted below:

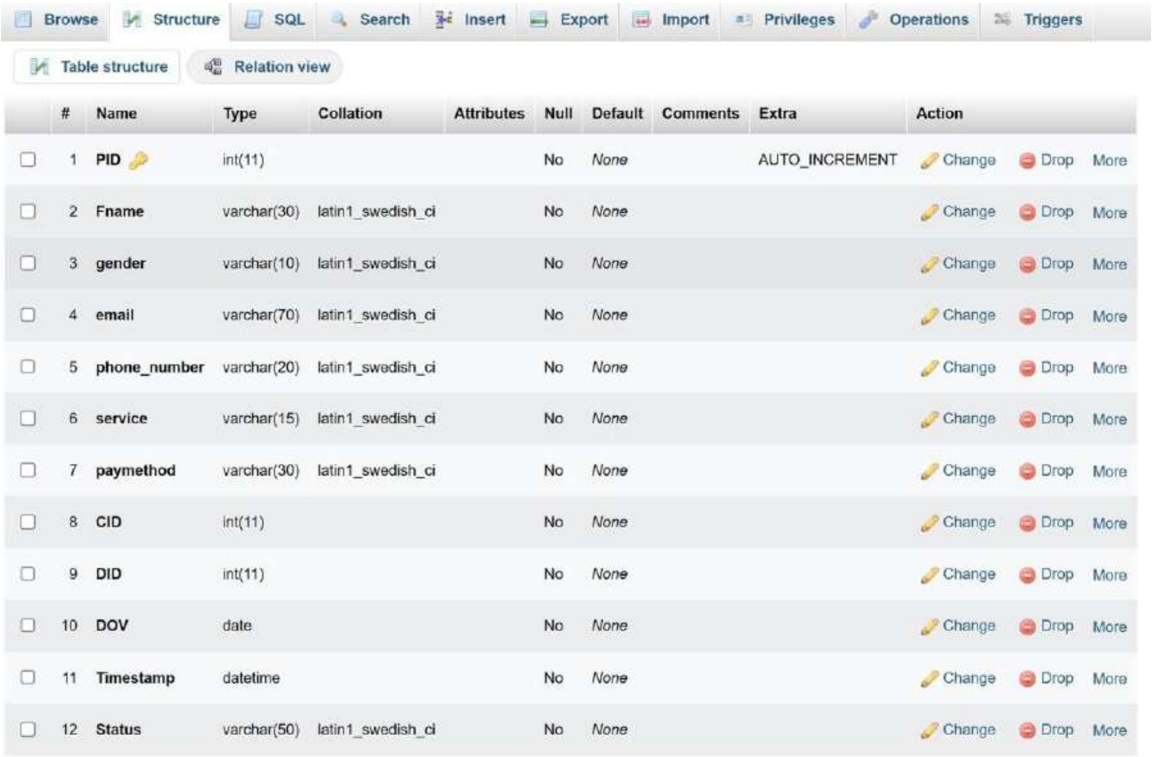
1. **Admin Table**: Contains all the information about admin and user as shown in Figure 2.1.



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	id	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
□	2	username	varchar(100)	utf8mb4_general_ci		No	*None*			Change Drop More
□	3	password	varchar(100)	utf8mb4_general_ci		No	*None*			Change Drop More

Figure 2.1: Admin table

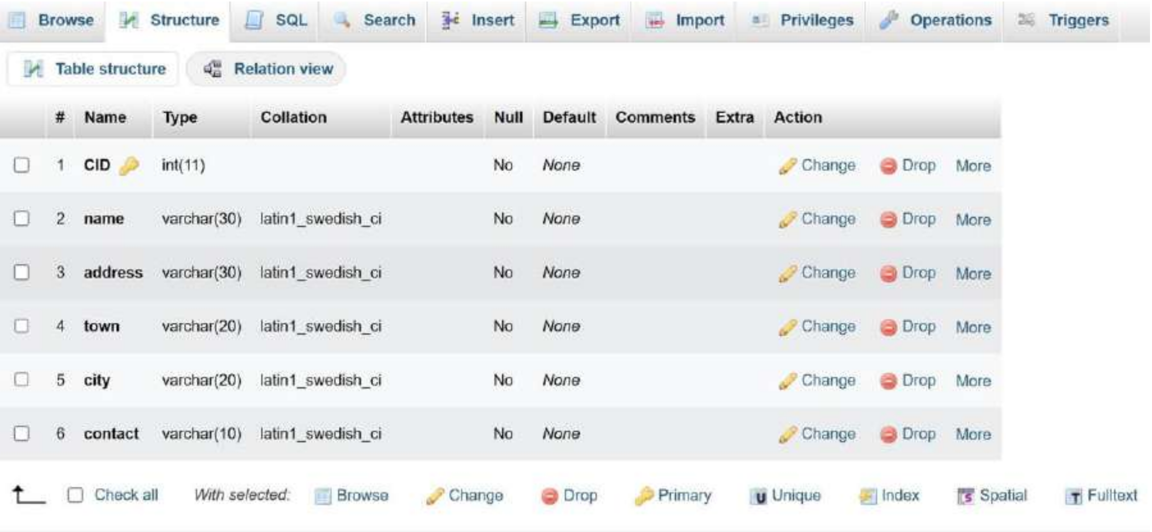
2. **Booking Table:** This table (see Figure 2.2) Contains information about booking schedule.



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	PID	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
□	2	Fname	varchar(30)	latin1_swedish_ci		No	*None*			Change Drop More
□	3	gender	varchar(10)	latin1_swedish_ci		No	*None*			Change Drop More
□	4	email	varchar(70)	latin1_swedish_ci		No	*None*			Change Drop More
□	5	phone_number	varchar(20)	latin1_swedish_ci		No	*None*			Change Drop More
□	6	service	varchar(15)	latin1_swedish_ci		No	*None*			Change Drop More
□	7	paymethod	varchar(30)	latin1_swedish_ci		No	*None*			Change Drop More
□	8	CID	int(11)			No	*None*			Change Drop More
□	9	DID	int(11)			No	*None*			Change Drop More
□	10	DOV	date			No	*None*			Change Drop More
□	11	Timestamp	datetime			No	*None*			Change Drop More
□	12	Status	varchar(50)	latin1_swedish_ci		No	*None*			Change Drop More

Figure 2.2: Booking table

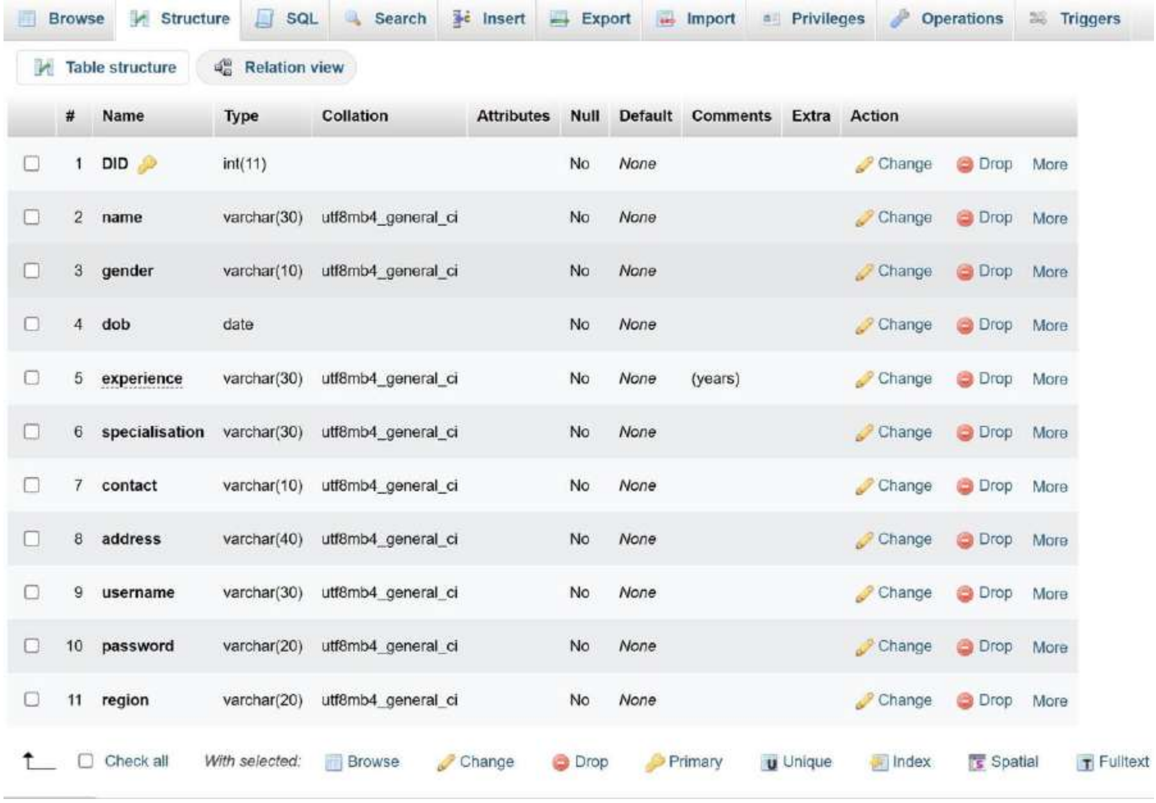
3. **Clinic table**: It contains information about the clinic (see Figure 2.3).



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
□	1	CID	int(11)			No	None			Change	Drop	More
□	2	name	varchar(30)	latin1_swedish_ci		No	None			Change	Drop	More
□	3	address	varchar(30)	latin1_swedish_ci		No	None			Change	Drop	More
□	4	town	varchar(20)	latin1_swedish_ci		No	None			Change	Drop	More
□	5	city	varchar(20)	latin1_swedish_ci		No	None			Change	Drop	More
□	6	contact	varchar(10)	latin1_swedish_ci		No	None			Change	Drop	More

Figure 2.3: order track history table

4. **Doctors table**: Contains information about doctors such as (Name, Job, Places, and Social media ……..) as illustrated in Figure 2.4 .



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
□	1	DID	int(11)			No	None			Change	Drop	More
□	2	name	varchar(30)	utf8mb4_general_ci		No	None			Change	Drop	More
□	3	gender	varchar(10)	utf8mb4_general_ci		No	None			Change	Drop	More
□	4	dob	date			No	None			Change	Drop	More
□	5	experience	varchar(30)	utf8mb4_general_ci		No	None	(years)		Change	Drop	More
□	6	specialisation	varchar(30)	utf8mb4_general_ci		No	None			Change	Drop	More
□	7	contact	varchar(10)	utf8mb4_general_ci		No	None			Change	Drop	More
□	8	address	varchar(40)	utf8mb4_general_ci		No	None			Change	Drop	More
□	9	username	varchar(30)	utf8mb4_general_ci		No	None			Change	Drop	More
□	10	password	varchar(20)	utf8mb4_general_ci		No	None			Change	Drop	More
□	11	region	varchar(20)	utf8mb4_general_ci		No	None			Change	Drop	More

Figure 2.4: product table

3.5The Activity Diagram

A Activity diagram (AD) are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e., workflows), as well as the data flows intersecting with the related activities as shown in Figure 2.5. Although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of data between activities through one or more data stores .We have a website where can search about doctors and we can view so after we can decide to be choose the doctor view and clinic place … another important is browser we can pay easily from the server and that make the booking more easily.

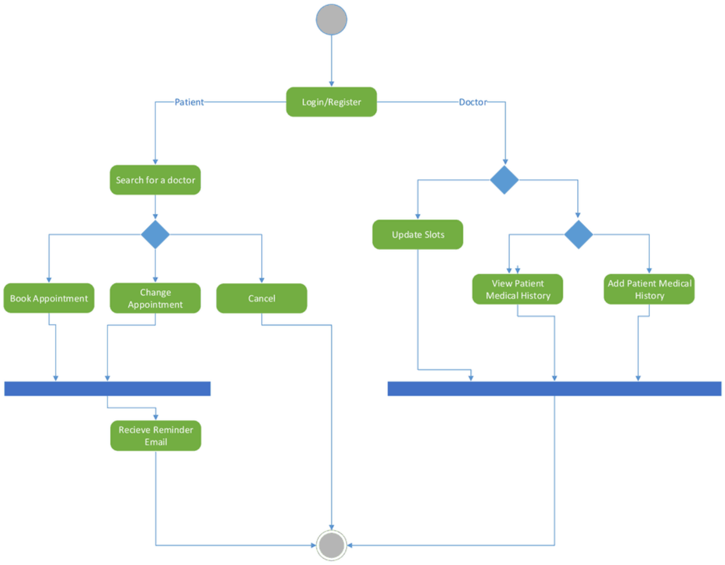


Figure 2.5 : The Activity diagram

3.6 Implementation Tools

An implementation tool is a collection of tools that, when used together or separately, can be useful for putting into practice a new initiative, project, practice, or program. The toolkit can be used on its whole, although users may find that some sections are very instructive for their requirements. We have used both of our sections, software, and hardware, in our project to administer the system, as any project that is built must contain both a software tool and a hardware tool.

3.6.1 Hardware Tools

We utilized a laptop with the specs of a Cori 5, G10, Ram 8, and a 128SSD hard drive firm MacBook Pro to arrange and carry out our work so that we could design our website in an advanced fashion. This was a tremendous aid in terms of hardware and made it easier for us to use computers technically. This is why the codes we created with the aid of the laptop enable us to view the fruits of our labor from the perspective of a victor. As a result, our laptop, which we utilized to conduct research for our website, was a huge assistance overall in assisting us in making our website function better.

3.6.2 Software Tools

An extremely quick source code editor is included with Visual Studio Code (VS) and is perfect for regular use. VS Code's support for hundreds of languages, syntax highlighting, bracket matching, auto-indentation, box selection, and snippets on my website helps us be more productive quickly. Software tools often have two components, the front end, and the back end.

1 . Front End:

- The front end of software refers to the user-facing part of an application, which is designed to interact with the user and present information in a visually appealing and intuitive way. This includes the user interface (UI) and user experience (UX) components, such as buttons, forms, menus, icons, and animations.
- Front-end development involves using languages such as HTML, CSS, and JavaScript to create and style the UI, and to implement interactions and functionality that allow users to navigate the application, input data, and interact with the back-end of the software. The front-end of software is what the user sees and interacts with directly, and a well-designed front-end can greatly enhance the user experience of an application.

- **HTML**

- HTML stands for Hypertext Markup Language. It is a markup language used for creating and structuring content on the web. HTML provides a standardized way to create web pages that can be viewed in web browsers.
- HTML consists of a set of tags that define the structure and layout of web content. These tags are used to mark up text, images, videos, forms, and other types of content to make them readable by web browsers. HTML tags are typically written using angle brackets, such as <html>, <body>, <p>, <img>, and <a>.
- HTML is a foundational technology for web development, and is often used in combination with other technologies such as CSS and JavaScript to create interactive and visually appealing web pages. It is constantly evolving, with new versions and features being added regularly to keep up with the changing needs of the web.

- **CSS**

- CSS stands for Cascading Style Sheets. It is a language used to describe the visual style and layout of HTML documents. CSS allows web developers to separate the presentation of a web page from its content, making it easier to maintain and update the design of a website.
- CSS works by using selectors to target HTML elements and apply specific styles to them. Styles can be applied to elements such as text, images, backgrounds, and layout properties such as margins and padding. CSS styles can be defined in an external style sheet, embedded within the HTML document, or applied inline to specific elements.
- CSS provides a wide range of styling options, including color, font, text size and spacing, layout and positioning, and animation effects. It also supports responsive design, allowing web developers to create layouts that adapt to different screen sizes and devices.
- CSS is an essential part of modern web development, and is often used in combination with HTML and JavaScript to create complex and dynamic web applications.

- **JavaScript**

- JavaScript is a high-level, dynamic programming language that is commonly used in web development to create interactive user interfaces and dynamic web pages. JavaScript can also be used on the server-side with technologies such as Node.js.
- JavaScript allows developers to add behavior to web pages, making them more responsive to user input and enabling dynamic updates to the page without requiring a full reload. JavaScript can be used to create a wide range of interactive features, including forms, menus, sliders, pop-ups, and animations.
- JavaScript is often used in combination with HTML and CSS to create full-featured web applications. It can be used to access and modify the content of an HTML page, and to interact with web APIs to fetch data and perform operations on the server-side.
- JavaScript is a versatile language that is constantly evolving, with new features and updates being added regularly. It is widely supported by modern web browsers and is a key technology for creating modern web applications.

- **Bootstrap**
- Bootstrap is a popular open-source front-end development framework used for designing responsive and mobile-first websites and web applications. It was created by Twitter and is now maintained by a community of developers.
- Bootstrap includes a collection of HTML, CSS, and JavaScript components and tools, such as typography, forms, buttons, navigation bars, modals, and carousels. These components are designed to work together and can be easily customized to create modern and visually appealing user interfaces.
- One of the main advantages of using Bootstrap is its grid system, which allows developers to create flexible and responsive layouts that adapt to different screen sizes and devices. Bootstrap also includes a set of pre-designed templates and themes that can be used as a starting point for building a new website or application.
- Bootstrap is widely used in web development because of its ease of use, cross-browser compatibility, and responsive design capabilities. It can save developers a lot of time and effort in building complex UIs, as many common components are already included and can be easily customized to fit the specific needs of a project.

2.BackEnd :

- The backend of software refers to the part of an application that handles the behind-the-scenes logic, data processing, and storage. It is responsible for managing and processing data, responding to requests from the front-end, and interacting with databases, APIs, and other external systems.
- Backend software is typically built using programming languages such as Java, Python, Ruby, PHP, and Node.js. It often involves the use of frameworks and libraries such as Django, Flask, Ruby on Rails, Laravel, and Express.js.
- The backend of an application is responsible for managing and storing data securely, as well as handling user authentication and authorization. It may also involve the use of caching mechanisms, load balancers, and other technologies to optimize performance and scalability.
- In summary, the backend of software is the part of an application that is responsible for handling data processing, storage, and management, as well as interacting with other systems to provide functionality to the front-end.

- **PHP**

- PHP (Hypertext Preprocessor) is a server-side programming language designed for web development. It is an open-source language widely used for creating dynamic websites and web applications. PHP code is embedded in HTML code and is executed on the server-side, generating dynamic content that is sent to the client's browser as plain HTML.
- PHP is a popular choice for web development because of its ease of use, wide support, and large community. It can be used to create a wide range of web applications, from simple scripts to complex content management systems and e-commerce platforms.
- PHP has a rich set of features that enable developers to create dynamic, interactive web applications. These features include support for databases, forms, cookies, session management, and many more. PHP also has a large number of third-party libraries and frameworks, such as Laravel, CodeIgniter, and Symfony, which can help developers streamline development and build more complex applications.
- In summary, PHP is a server-side programming language used for web development, widely popular due to its ease of use, support, and community. It has a rich set of features and is commonly used for creating dynamic, interactive web applications.

- ## Apache

- Apache is an open-source web server software designed for hosting websites and serving web pages. It is the most widely used web server software on the internet, powering over 40% of all websites.
- Apache is known for its stability, flexibility, and performance. It runs on a variety of operating systems, including Windows, Linux, macOS, and Unix. Apache is modular and extensible, allowing developers to add new functionality to the web server using modules.
- Apache supports a wide range of features and technologies, such as SSL/TLS encryption, virtual hosting, URL rewriting, access control, and logging. It can also be configured to work with various programming languages, such as PHP, Perl, Python, and Ruby, and supports various databases, such as MySQL and PostgreSQL.
- Apache is often used in combination with other open-source software, such as PHP and MySQL, to create a full-stack web development environment. It is also commonly used in enterprise environments for hosting and serving internal and external web applications.
- In summary, Apache is an open-source web server software widely used for hosting websites and serving web pages. It is known for its stability, flexibility, and performance and supports a wide range of features, technologies, and programming languages.

- ## XAMPP

- XAMPP is a free, open-source, cross-platform web server software package that includes the Apache web server, MySQL database, PHP, and Perl. The name XAMPP stands for cross-platform (X), Apache, MySQL, PHP, and Perl.
- XAMPP is designed to provide an easy-to-install and easy-to-use web server environment for development and testing purposes. It is often used by developers to create and test dynamic web applications on their local machines before deploying them to a live server. XAMPP can be installed on various operating systems such as Windows, Linux, and macOS.
- In addition to Apache, MySQL, PHP, and Perl, XAMPP includes other useful tools such as phpMyAdmin, which is a web-based database administration tool, and FileZilla, which is an FTP client for transferring files to and from the server.
- XAMPP simplifies the process of setting up a local web server environment, allowing developers to quickly create and test web applications without having to worry about configuring separate components individually. It is an ideal tool for beginners to learn web

development or for experienced developers to create and test web applications in a development environment.

- In summary, XAMPP is a free, open-source, cross-platform web server software package that includes Apache, MySQL, PHP, and Perl, designed to provide an easy-to-install and easy-to-use web server environment for development and testing purposes.

- ## MySQL

- MySQL is a free, open-source, relational database management system (RDBMS) that is widely used for web development. It is one of the most popular databases in use, with millions of installations worldwide.
- MySQL allows users to create, read, update, and delete data stored in a relational database. It supports SQL (Structured Query Language), which is a standard language used to manage and manipulate relational databases.
- MySQL is known for its scalability, reliability, and ease of use. It can handle large amounts of data, making it ideal for web applications that need to store and manage large volumes of information. It is also highly configurable and can be optimized for different workloads, making it suitable for a wide range of use cases.
- MySQL can be used with various programming languages, including PHP, Python, and Java. It also includes a command-line tool and a graphical user interface (GUI) for managing databases and tables.
- In addition to being free and open-source, MySQL also offers a paid, commercial version called MySQL Enterprise, which includes additional features and support services.
- In summary, MySQL is a free, open-source, relational database management system widely used for web development. It supports SQL and is known for its scalability, reliability, and ease of use. It can be used with various programming languages and includes both a command-line tool and a graphical user interface for managing databases and tables.

Chapter Four

The Implemented System, Tests and Experimental Outcomes

4. The Implemented System, Tests and Experimental Outcomes

4.1 Introduction

Tthe aim of this chapter is to describe the implementation if the design system, which was given in details in chapter three. The system requirements and the implementation toolset are discussed also. The system is presented and the interaction of user with the implemented system through the demonstration as test are discussed. Finally, the empirical outcomes are discussed.

4.2 Experimental Outcomes

The problems that are encountered with the traditional methods concerning the Backend pages, Frontend Pages, and database are noticed during the implementation of system and solved as the following:

4.3 Experimental Outcomes with Screen-shots

The problems that are encountered with the traditional methods concerning the Backend pages, Frontend Pages, and database are noticed during the implementation of system and solved as the following

1 - Backend pages.

-Login Page

This is the login of our backend website, for patient's operation the user or administration Must have their username and password to enter the backend for operation as shown in Figure 4.1.

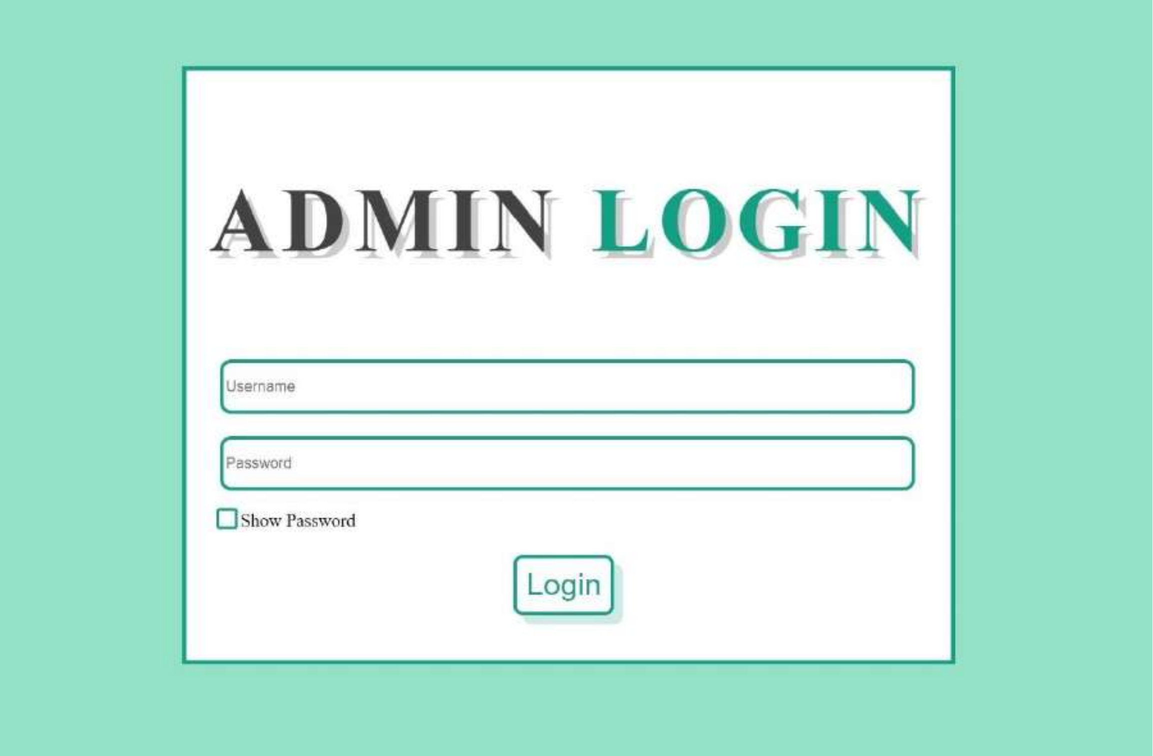
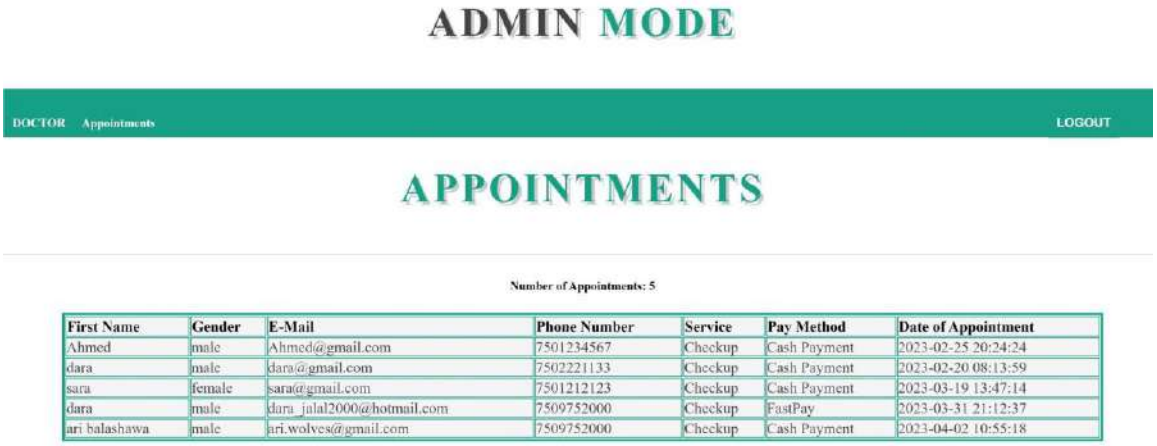


Figure 4.1 Admin Login Page

- When **Admin** Enters, He controls all activities of the website such as updating order status, manage users, create update or delete doctor, table, time, create and update and delete and manage the scale of products, view user login logs.
- It can also see and manage the time that the patients took appointment, from which doctors and day that appointed. Figure 4.2 illustrate the admin mode.



APPOINTMENTS

Number of Appointments: 5

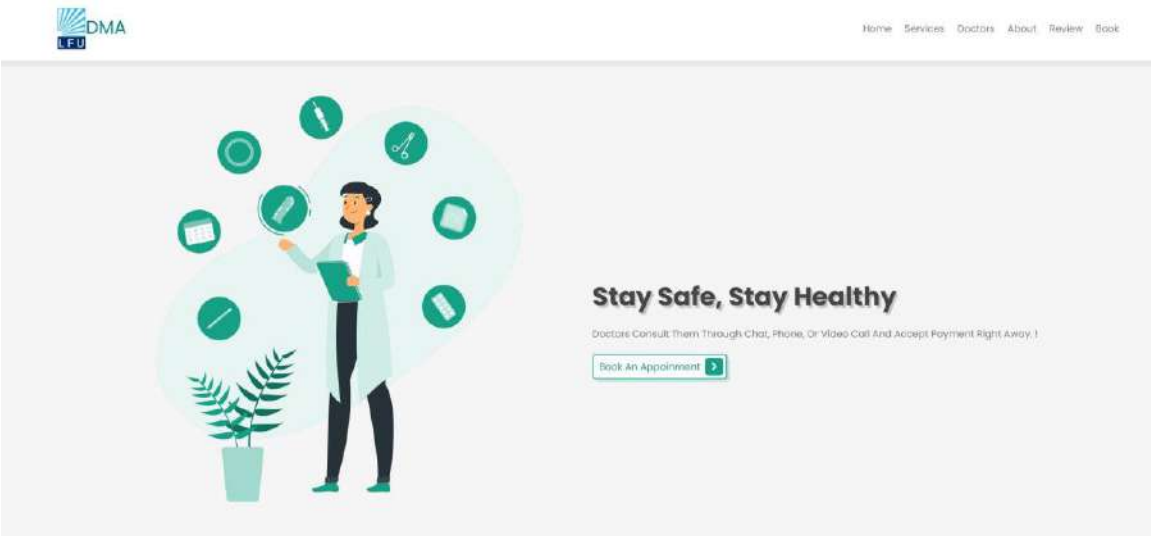
First Name	Gender	E-Mail	Phone Number	Service	Pay Method	Date of Appointment
Ahmed	male	Ahmed@gmail.com	7501234567	Checkup	Cash Payment	2023-02-25 20:24:24
dara	male	dara@gmail.com	7502221133	Checkup	Cash Payment	2023-02-20 08:13:59
sara	female	sara@gmail.com	7501212123	Checkup	Cash Payment	2023-03-19 13:47:14
dara	male	dara_jalal2000@hotmail.com	7509752000	Checkup	FastPay	2023-03-31 21:12:37
ari balashawa	male	ari.wolves@gmail.com	7509752000	Checkup	Cash Payment	2023-04-02 10:55:18

-

Figure 4.2 Admin Mode

2- Frontend Pages.

All(main) Page: this page contains all the pages which you can access such as the categories, the login page, Services, Doctors, About Us, and Book Appointment as presented in Figure 4.3.

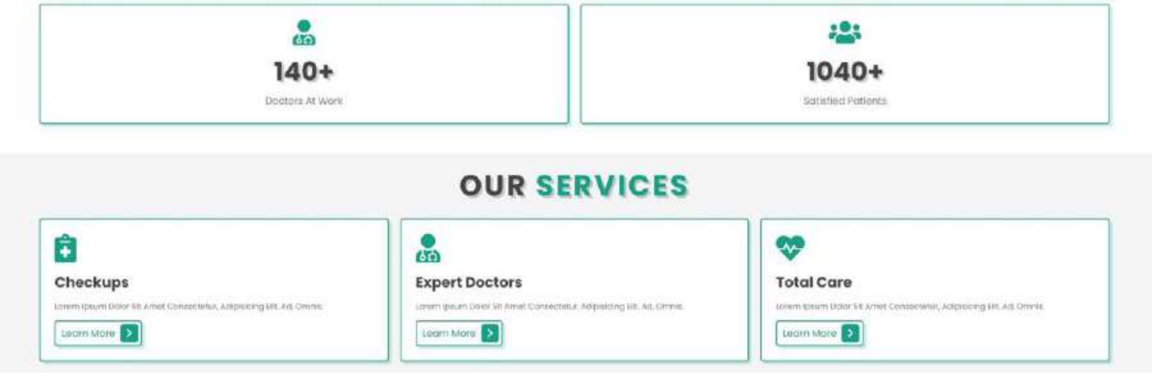


Stay Safe, Stay Healthy

Doctors Consult Them Through Chat, Phone, Or Video Call And Accept Payment Right Away. !

Figure 4.3 Home page

"OUR SERVICES" is a phrase commonly used on websites to describe the various services offered by the work place that the website has been created as showed in Figure 4.4.


140+

Doctors At Work

1040+

Satisfied Patients

OUR SERVICES

Checkups

Lorem Ipsum Dolor Sit Amet Consectetur, Adipisicing Elit. Ad, Omnis.

Learn More

Expert Doctors

Lorem Ipsum Dolor Sit Amet Consectetur, Adipisicing Elit. Ad, Omnis.

Learn More

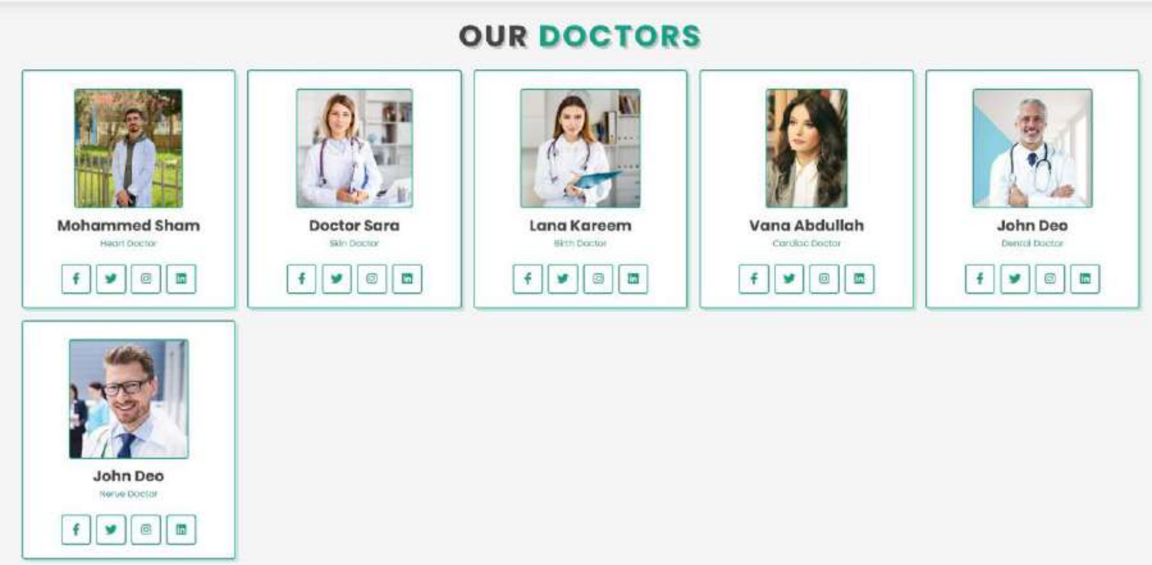
Total Care

Lorem Ipsum Dolor Sit Amet Consectetur, Adipisicing Elit. Ad, Omnis.

Learn More

Figure 4.4 Our Services Page

"OUR doctors" could refer to a number of doctors, such as a medical clinic or hospital, a healthcare provider network, or a health insurance company, which they're available on the website. For each doctor have special jobs title and their social medias account which you can only click on them and contact with the doctor that are available on the website(see Figure 4.5).


OUR DOCTORS

Mohammed Sham

Heart Doctor

Doctor Sara

Skin Doctor

Lana Kareem

Birth Doctor

Vana Abdullah

Cardiac Doctor

John Deo

Dental Doctor

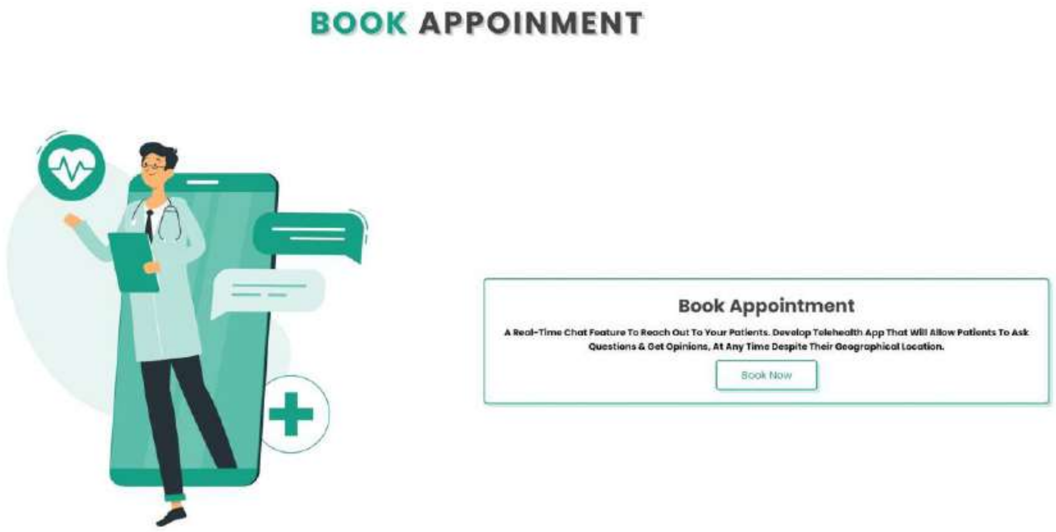
John Deo

Nerve Doctor

Figure 4.5 Our Doctors Page

A doctor booking page on a website is typically a feature that allows patients to schedule appointments with doctors or other healthcare providers online. This can be convenient for patients who prefer to book appointments outside of regular business hours or who do not have the time to call during office hours as displayed in Figure 4.6 and 4.7.

Doctor booking pages typically require patients to select the type of service they need, the location of the appointment, the date and time that works best for them, and their personal information, such as their name and contact details. Some doctor booking pages may also include a search function that allows patients to search for doctors based on their specialty or location.

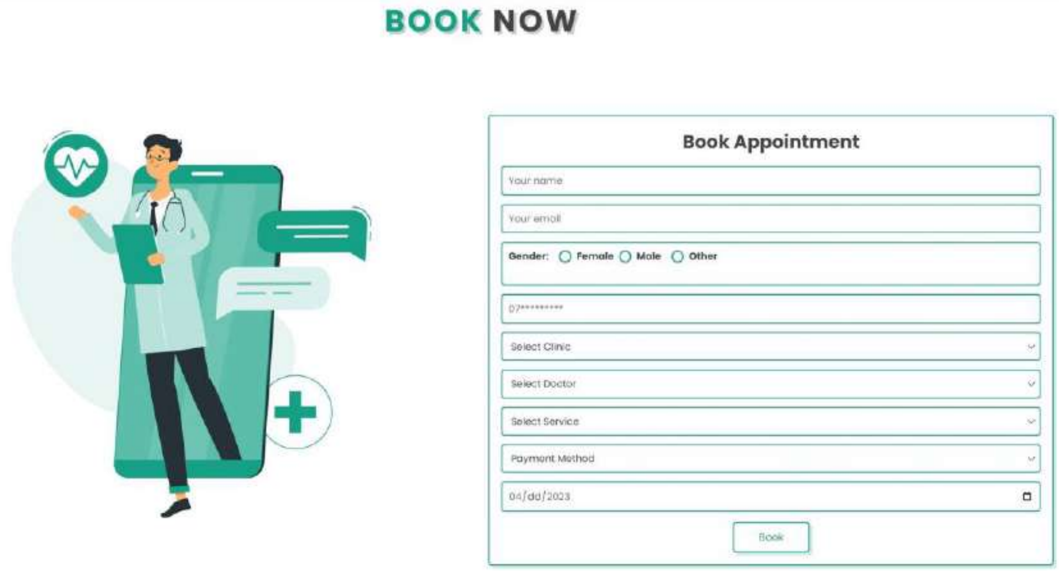


BOOK APPOINMENT

Book Appointment

A Real-Time Chat Feature To Reach Out To Your Patients. Develop Telehealth App That Will Allow Patients To Ask Questions & Get Opinions, At Any Time Despite Their Geographical Location.

Figure 4.6 Book Appointment Page



BOOK NOW

Book Appointment

Your name

Your email

Gender: ○ Female ○ Male ○ Other

07*********

Select Clinic

Select Doctor

Select Service

Payment Method

04/dd/2023

Figure 4.6 Book Appointment Page options.

Chapter Five
Conclusions, Limitations, and Future Works

Chapter Five
Conclusions, Limitations, and Future Works

5. Conclusions, Limitations, and Future Works

5.1 Conclusions

In conclusion, doctor appointment systems have become an essential part of modern healthcare services, as they enable patients to schedule appointments with healthcare professionals easily and efficiently. By reducing waiting times, improving scheduling processes, and reducing administrative burdens, doctor appointment systems can help to improve the overall quality of care for patients. Additionally, as healthcare systems continue to evolve, the development of effective doctor appointment systems will be critical to ensuring that patients receive the care they need in a timely and efficient manner. However, designing and implementing a website for reserving data and storing patient information is an essential component of modern healthcare. A well-designed website can improve healthcare outcomes, enhance patient satisfaction, and streamline healthcare processes. It is crucial to prioritize user accessibility, security, and compliance with relevant data protection and privacy regulations to establish trust between patients and healthcare providers. With careful planning and implementation, a website for reserving data and storing patient information can transform the delivery of healthcare services. the implementation of a doctor appointment system online can bring numerous benefits for both patients and healthcare providers. By offering a user-friendly interface, mobile compatibility, and customizable booking options, patients can easily schedule appointments from anywhere and at any time, without experiencing long wait times or the frustration of traditional appointment booking systems.

Overall, the development and implementation of a doctor appointment system online can bring significant benefits for patients, healthcare providers, and the healthcare system as a whole. It can help to streamline operations, improve patient experiences, and ultimately, enhance the quality of care provided to patients.

5.2 Recommendations for Future Works

There are several suggestions for future works:

1. Integration with Telemedicine: As more patients turn to telemedicine for remote consultations, the integration of telemedicine into the doctor appointment system can provide a comprehensive and convenient solution for patients.

2. Patient Feedback: Gathering feedback from patients can help healthcare providers identify areas for improvement and tailor the booking system to meet the specific needs and preferences of patients.
3. Integration with Wearable Devices: Integrating wearable devices, such as fitness trackers or smartwatches, can provide healthcare providers with real-time data on patients' health metrics.

By implementing these future works, healthcare providers can continue to improve the doctor appointment system, providing patients with convenient, efficient, and personalized care.

5.3Limitations

The researchers suffer from many limitations during the design and implementation system, as listed in the following:

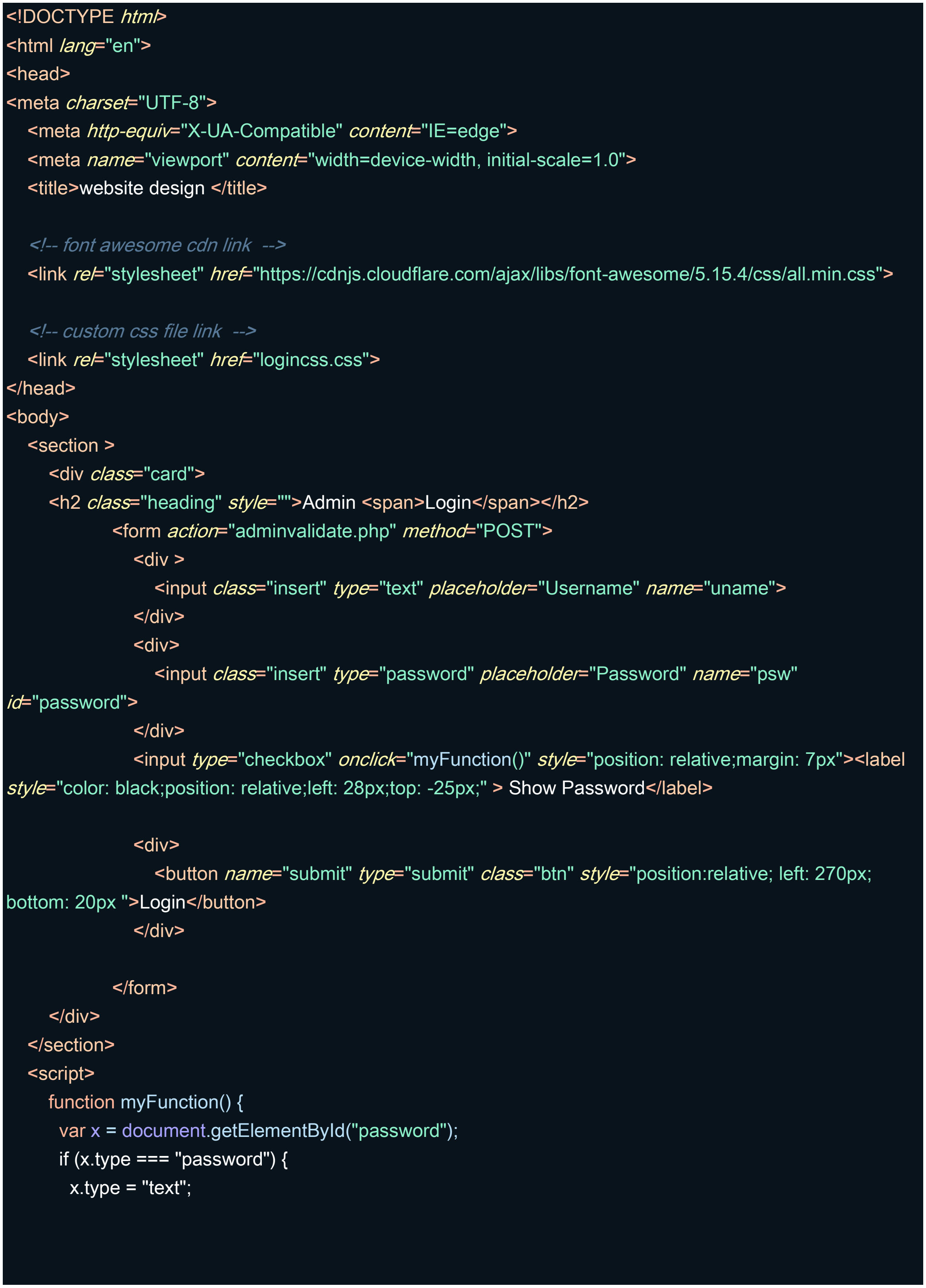
1. Technical Issues: Like any technological solution, doctor appointment systems may experience technical issues, such as system crashes, slow loading times, or software bugs.
2. Internet Connectivity: Doctor appointment systems are dependent on stable and reliable internet connectivity.
3. User Resistance: Some patients may be resistant to using an online booking system, preferring traditional methods such as phone calls or in-person appointments.
4. Cost: Implementing a doctor appointment system may require significant upfront costs, including hardware and software purchases, system integration, and staff training.

References

1. Farahani, M.R., Rajabzadeh, A. and Esmaeilzadeh, P., 2021. Developing a cloud-based intelligent appointment scheduling system for healthcare services. Computers in Industry, 128, p.103454.
2. Khalid, S., Asghar, A., Arshad, J., Raza, S. and Muhammad, S., 2020. Design and Implementation of a Web-Based Appointment System for Small Healthcare Clinics. In 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 546-550). IEEE.
3. Shukla, P., Shukla, R. and Mehta, N.K., 2019. Doctor appointment system using the internet of things (IoT). In Proceedings of the 5th International Conference on Computing Communication and Automation (ICCCA 2019) (pp. 424-428). Springer.
4. Sood, S., Gupta, S., Ahuja, V. and Kaur, G., 2020. Development of a smart doctor appointment system using machine learning algorithms. Journal of Healthcare Engineering, 2020.
5. Software Advice. (2021). Patient Scheduling Software Buyer Report - 2021. [online] Available at: https://dataintelo.com/report/patient-scheduling-software-market [Accessed 9 Apr. 2023].
6. Zhao P, Yoo I, Lavoie J, Lavoie BJ, Simoes E. Web-Based Medical Appointment Systems: A Systematic Review. J Med Internet Res. 2017 Apr 26;19(4):e134. doi: 10.2196/jmir.6747. PMID: 28446422; PMCID: PMC5425771.
7. Zhang X, Yu P, Yan J. Patients' adoption of the e-appointment scheduling service: a case study in primary healthcare. Stud Health Technol Inform. 2014;204:176–181S
8. Zhang X, Yu P, Yan J, Hu H, Gouvei N. Patients' perceptions of web self-service applications in primary healthcare. Stud Health Technol Inform. 2012;178:242–249.
9. Karthikraj H, and et. al. Doctor Appointment System Using Cloud. Journal of Healthcare Engineering, Special Issue of Second International Conference on Advancements in Research and Development (ICARD 2021). Volume 03 Issue 03S March 2021.
10. Wang and Gupta, Adaptive Appointment Systems with Patient Preferences, Manufacturing & Service Operations Management, 2021, DOI: 10.1287/msom.1110.0332.
11. Vermeulen, et.al., Optimization of Online Patient Scheduling with Urgencies and Preferences, Conference: Artificial Intelligence in Medicine, 12th Conference on Artificial Intelligence in Medicine, Italy.
12. Zhao P, Yoo I, Lavoie J, Lavoie B, Simoes E Web-Based Medical Appointment Systems: A Systematic ReviewJ Med Internet Res 2017;19(4):e134URL: https://www.jmir.org/2017/4/e134,DOI: 10.2196/jmir.6747.
13. Software Advice. (2021). https://www.siemens-healthineers.com/services/value-partnerships/portfolio-modules/digital-platforms [Accessed 9 Apr. 2023].

Appendix

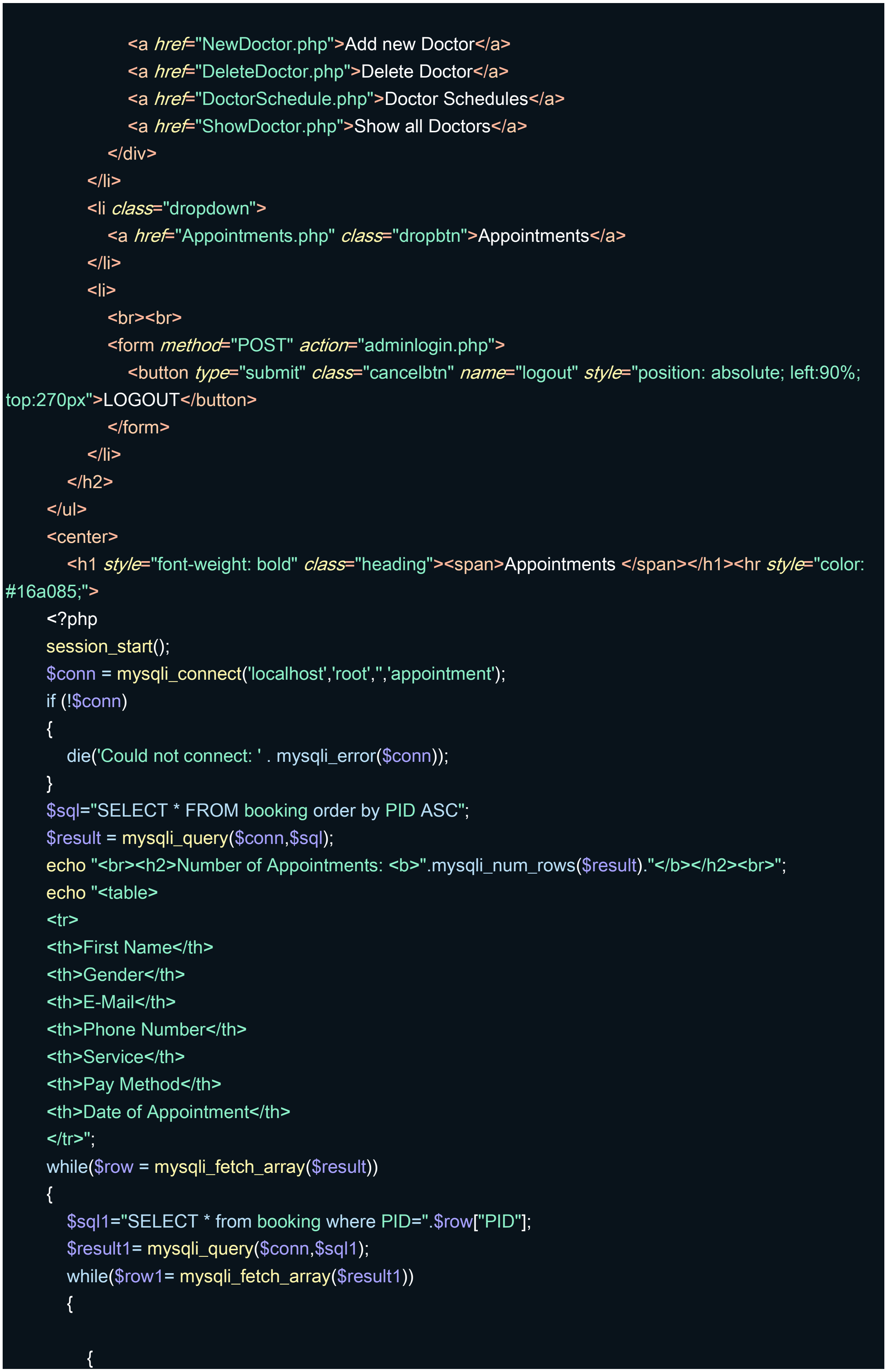
Appendix: Backend



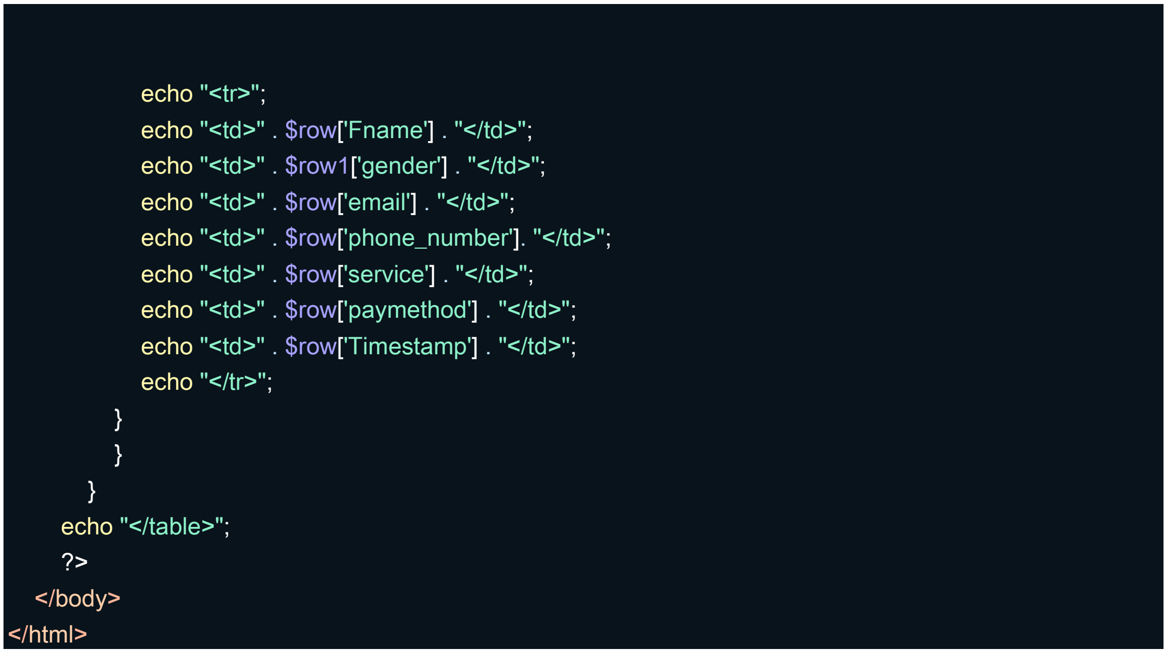
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>website design </title>

    <!-- font awesome cdn link  -->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css">

    <!-- custom css file link  -->
    <link rel="stylesheet" href="logincss.css">
</head>
<body>
    <section >
        <div class="card">
        <h2 class="heading" style="">Admin <span>Login</span></h2>
                    <form action="adminvalidate.php" method="POST">
                        <div >
                            <input class="insert" type="text" placeholder="Username" name="uname">
                        </div>
                        <div>
                            <input class="insert" type="password" placeholder="Password" name="psw"
id="password">
                        </div>
                        <input type="checkbox" onclick="myFunction()" style="position: relative;margin: 7px"><label
style="color: black;position: relative;left: 28px;top: -25px;" > Show Password</label>

                        <div>
                            <button name="submit" type="submit" class="btn" style="position:relative; left: 270px;
bottom: 20px ">Login</button>
                        </div>

                    </form>
        </div>
    </section>
    <script>
        function myFunction() {
          var x = document.getElementById("password");
          if (x.type === "password") {
            x.type = "text";
        } else {
            x.type = "password";
        }
    }
</script>
</body>
</html>
```

Appointments.php: Figure 3.2(Admin Mode)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Appointments</title>

        <link rel="stylesheet" href="ahome.css">
        <style>
    table
    {
        width:90%;
        border-collapse:separate;
        border: 5px solid var(--green);
        font-size: 30px;
    }
    th
    {
        border: 2px solid var(--green);
        color: black;
        text-align: left;
    }
    tr,td
    {
        border: 2px solid var(--green);
        background-color: whitesmoke;
        color:#444;
        padding: 0px;
    }
        </style>
    </head>
    <body>
    <h1 class="heading">ADMIN <span>MODE</span></h1>
        <ul>
            <br>
            <h2>
             <li class="dropdown">
                    <a class="dropbtn">DOCTOR</a>
                    <div class="dropdown-content">
```

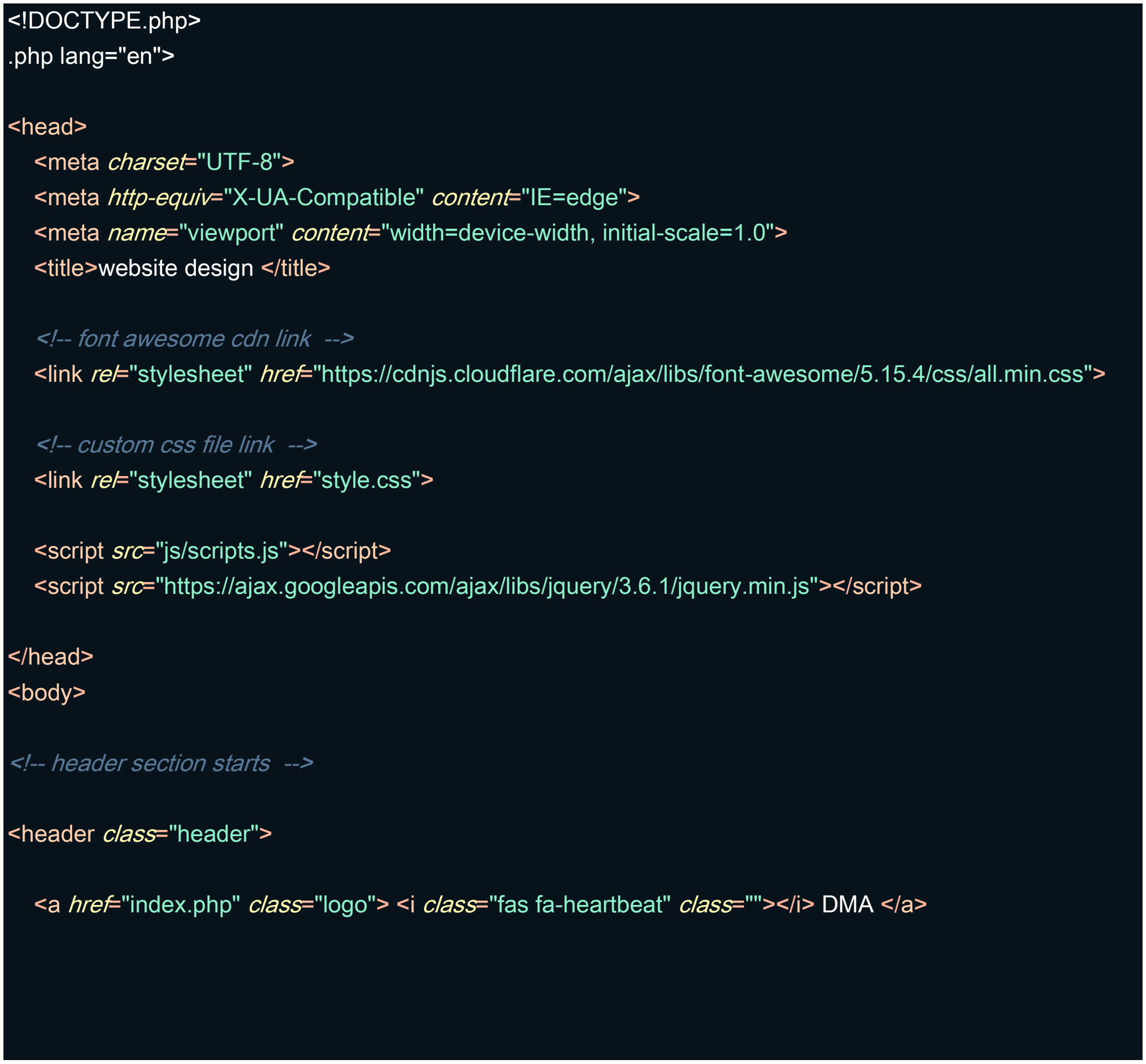


```
                    <a href="NewDoctor.php">Add new Doctor</a>
                    <a href="DeleteDoctor.php">Delete Doctor</a>
                    <a href="DoctorSchedule.php">Doctor Schedules</a>
                    <a href="ShowDoctor.php">Show all Doctors</a>
                </div>
            </li>
            <li class="dropdown">
                <a href="Appointments.php" class="dropbtn">Appointments</a>
            </li>
            <li>
                <br><br>
                <form method="POST" action="adminlogin.php">
                    <button type="submit" class="cancelbtn" name="logout" style="position: absolute; left:90%;
top:270px">LOGOUT</button>
                </form>
            </li>
        </h2>
    </ul>
    <center>
        <h1 style="font-weight: bold" class="heading"><span>Appointments </span></h1><hr style="color:
#16a085;">
    <?php
    session_start();
    $conn = mysqli_connect('localhost','root','','appointment');
    if (!$conn)
    {
        die('Could not connect: ' . mysqli_error($conn));
    }
    $sql="SELECT * FROM booking order by PID ASC";
    $result = mysqli_query($conn,$sql);
    echo "<br><h2>Number of Appointments: <b>".mysqli_num_rows($result)."</b></h2><br>";
    echo "<table>
    <tr>
    <th>First Name</th>
    <th>Gender</th>
    <th>E-Mail</th>
    <th>Phone Number</th>
    <th>Service</th>
    <th>Pay Method</th>
    <th>Date of Appointment</th>
    </tr>";
    while($row = mysqli_fetch_array($result))
    {
        $sql1="SELECT * from booking where PID=".$row["PID"];
        $result1= mysqli_query($conn,$sql1);
        while($row1= mysqli_fetch_array($result1))
        {

            {
```

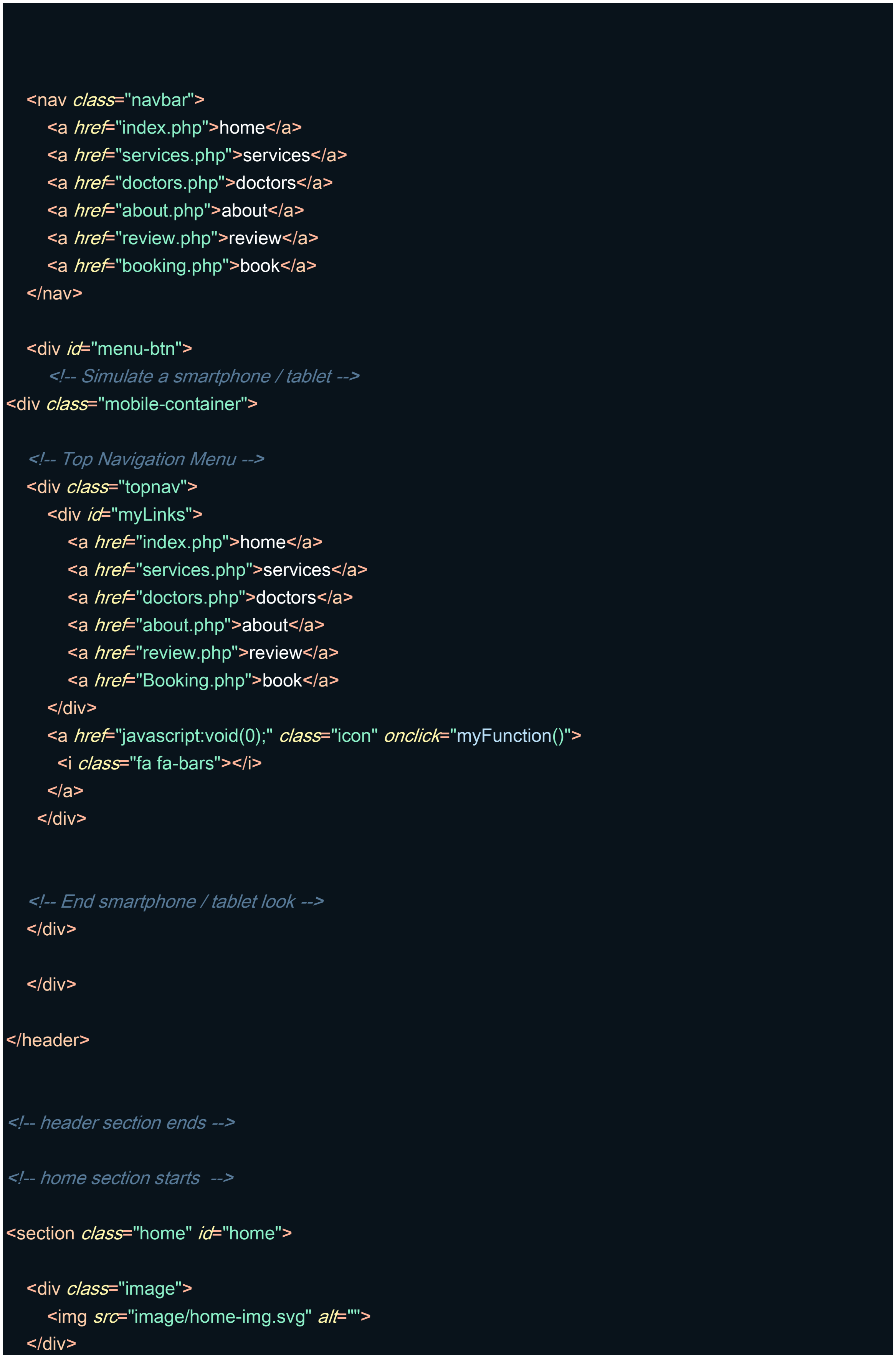


```
                echo "<tr>";
                echo "<td>" . $row['Fname'] . "</td>";
                echo "<td>" . $row1['gender'] . "</td>";
                echo "<td>" . $row['email'] . "</td>";
                echo "<td>" . $row['phone_number']. "</td>";
                echo "<td>" . $row['service'] . "</td>";
                echo "<td>" . $row['paymethod'] . "</td>";
                echo "<td>" . $row['Timestamp'] . "</td>";
                echo "</tr>";
            }
            }
        }
      echo "</table>";
      ?>
   </body>
</html>
```

Index.php: Figure 3.3

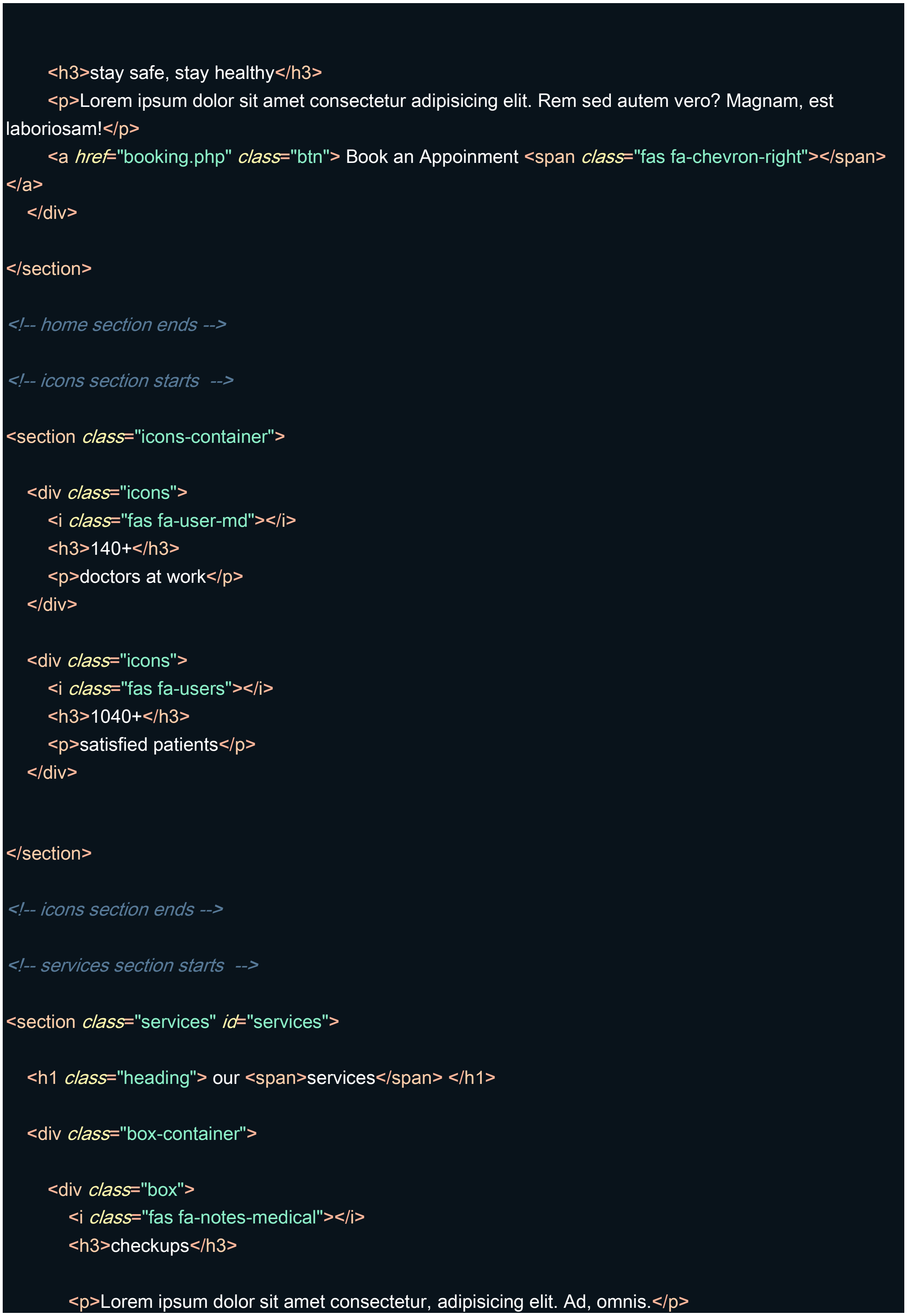


```
<!DOCTYPE.php>
.php lang="en">

<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>website design </title>

   <!-- font awesome cdn link  -->
   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css">

   <!-- custom css file link  -->
   <link rel="stylesheet" href="style.css">

   <script src="js/scripts.js"></script>
   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jquery.min.js"></script>

</head>
<body>

<!-- header section starts  -->

<header class="header">

   <a href="index.php" class="logo"> <i class="fas fa-heartbeat" class=""></i> DMA </a>
```



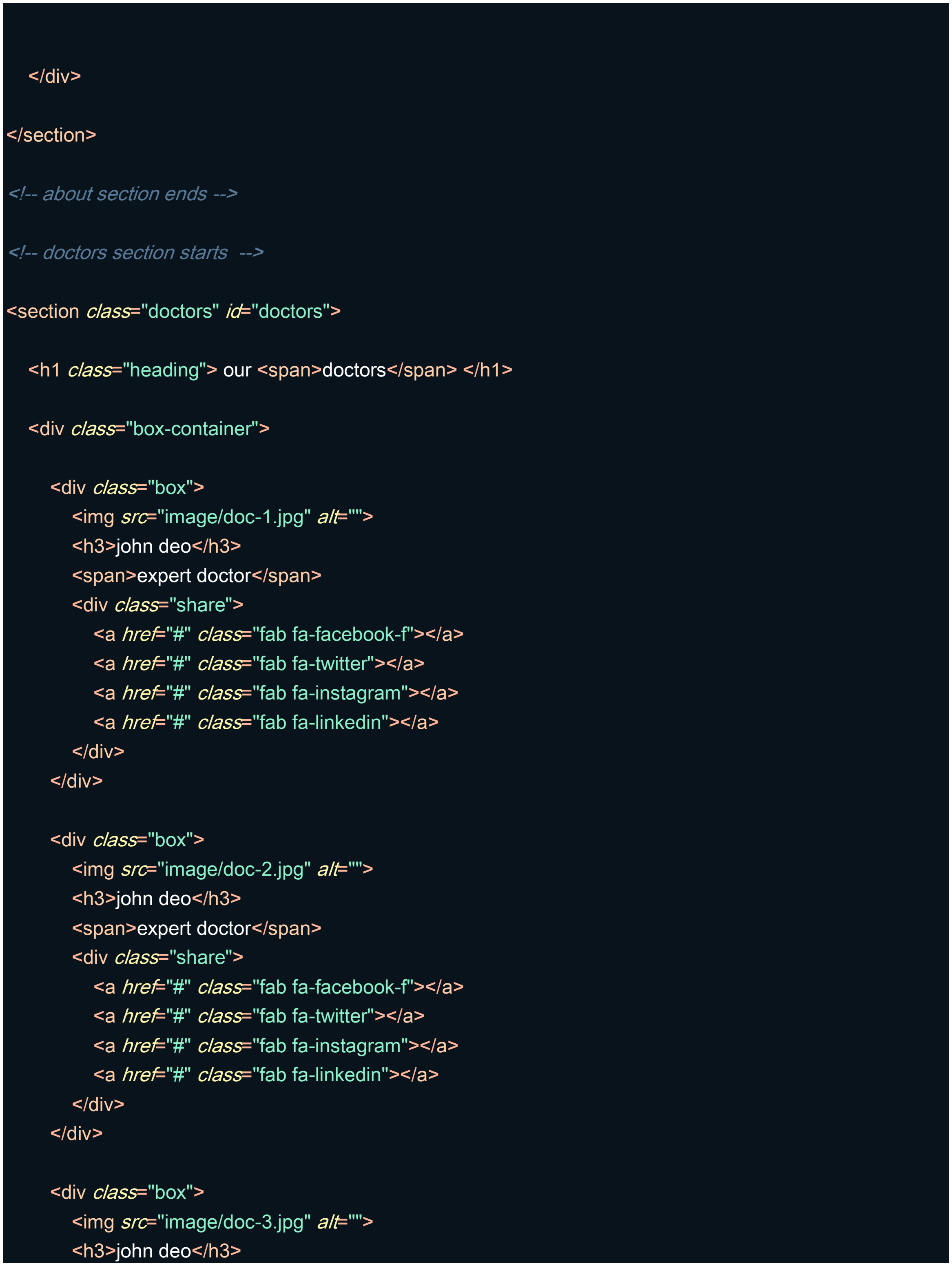
```
  <nav class="navbar">
    <a href="index.php">home</a>
    <a href="services.php">services</a>
    <a href="doctors.php">doctors</a>
    <a href="about.php">about</a>
    <a href="review.php">review</a>
    <a href="booking.php">book</a>
  </nav>

  <div id="menu-btn">
    <!-- Simulate a smartphone / tablet -->
<div class="mobile-container">

  <!-- Top Navigation Menu -->
  <div class="topnav">
    <div id="myLinks">
      <a href="index.php">home</a>
      <a href="services.php">services</a>
      <a href="doctors.php">doctors</a>
      <a href="about.php">about</a>
      <a href="review.php">review</a>
      <a href="Booking.php">book</a>
    </div>
    <a href="javascript:void(0);" class="icon" onclick="myFunction()">
     <i class="fa fa-bars"></i>
    </a>
   </div>


  <!-- End smartphone / tablet look -->
  </div>

  </div>

</header>


<!-- header section ends -->

<!-- home section starts  -->

<section class="home" id="home">

  <div class="image">
    <img src="image/home-img.svg" alt="">
  </div>
```

```
    <div class="content">
```



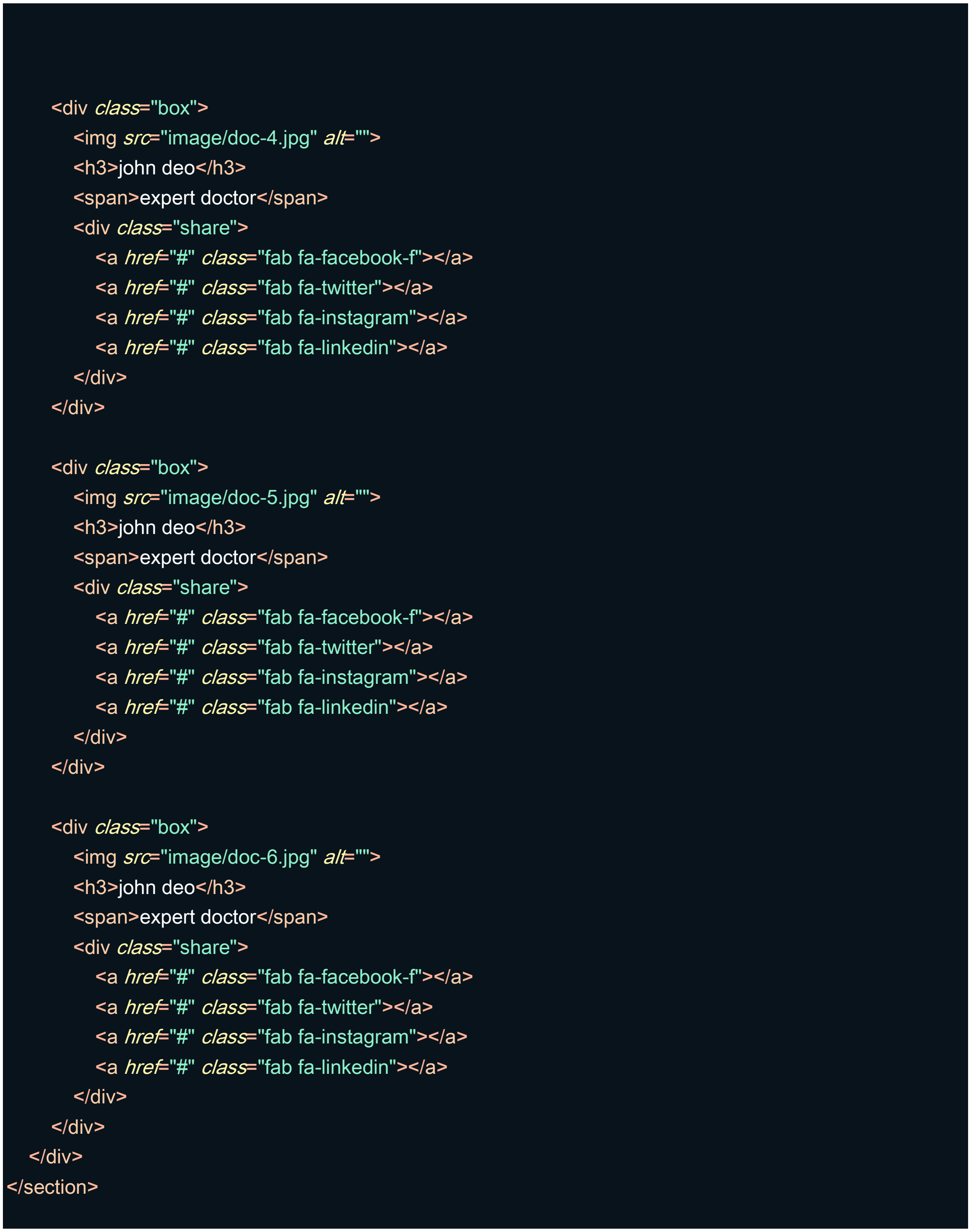
```
        <h3>stay safe, stay healthy</h3>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Rem sed autem vero? Magnam, est
laboriosam!</p>
        <a href="booking.php" class="btn"> Book an Appoinment <span class="fas fa-chevron-right"></span>
</a>
    </div>

</section>

<!-- home section ends -->

<!-- icons section starts  -->

<section class="icons-container">

    <div class="icons">
        <i class="fas fa-user-md"></i>
        <h3>140+</h3>
        <p>doctors at work</p>
    </div>

    <div class="icons">
        <i class="fas fa-users"></i>
        <h3>1040+</h3>
        <p>satisfied patients</p>
    </div>


</section>

<!-- icons section ends -->

<!-- services section starts  -->

<section class="services" id="services">

    <h1 class="heading"> our <span>services</span> </h1>

    <div class="box-container">

        <div class="box">
            <i class="fas fa-notes-medical"></i>
            <h3>checkups</h3>

            <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Ad, omnis.</p>
            <a href="booking.php" class="btn"> learn more <span class="fas fa-chevron-right"></span> </a>
        </div>
```

```
        <div class="box">
            <i class="fas fa-user-md"></i>
            <h3>expert doctors</h3>
            <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Ad, omnis.</p>
            <a href="doctors.php" class="btn"> learn more <span class="fas fa-chevron-right"></span> </a>
        </div>

        <div class="box">
            <i class="fas fa-heartbeat"></i>
            <h3>total care</h3>
            <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Ad, omnis.</p>
            <a href="services.php" class="btn"> learn more <span class="fas fa-chevron-right"></span> </a>
        </div>

    </div>

</section>

<!-- services section ends -->

<!-- about section starts  -->

<section class="about" id="about">

    <h1 class="heading"> <span>about</span> us </h1>

    <div class="row">

        <div class="image">
            <img src="image/about-img.svg" alt="">
        </div>

        <div class="content">
            <h3>we take care of your healthy life</h3>
            <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Iure ducimus, quod ex cupiditate ullam in assumenda maiores et culpa odit tempora ipsam qui, quisquam quis facere iste fuga, minus nesciunt.</p>
            <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Natus vero ipsam laborum porro voluptates voluptatibus a nihil temporibus deserunt vel?</p>
            <a href="about.php" class="btn"> learn more <span class="fas fa-chevron-right"></span> </a>
        </div>
```

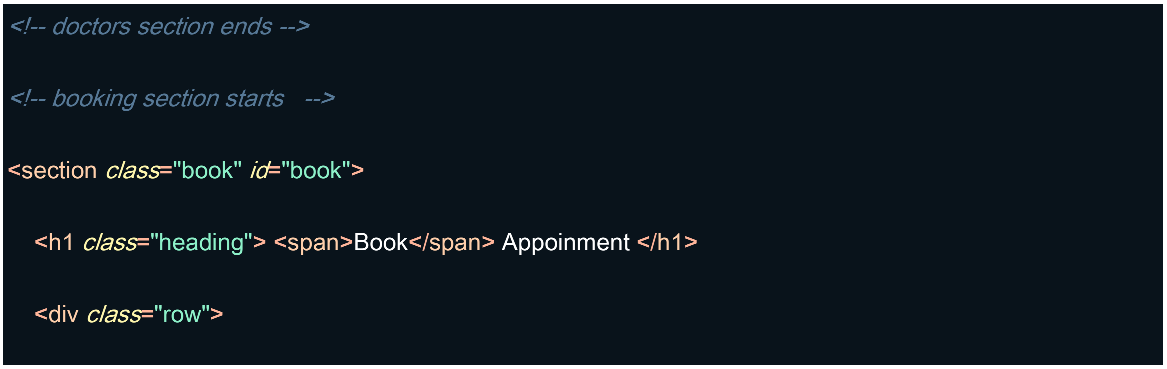


```
    </div>

</section>

<!-- about section ends -->

<!-- doctors section starts  -->

<section class="doctors" id="doctors">

    <h1 class="heading"> our <span>doctors</span> </h1>

    <div class="box-container">

        <div class="box">
            <img src="image/doc-1.jpg" alt="">
            <h3>john deo</h3>
            <span>expert doctor</span>
            <div class="share">
                <a href="#" class="fab fa-facebook-f"></a>
                <a href="#" class="fab fa-twitter"></a>
                <a href="#" class="fab fa-instagram"></a>
                <a href="#" class="fab fa-linkedin"></a>
            </div>
        </div>

        <div class="box">
            <img src="image/doc-2.jpg" alt="">
            <h3>john deo</h3>
            <span>expert doctor</span>
            <div class="share">
                <a href="#" class="fab fa-facebook-f"></a>
                <a href="#" class="fab fa-twitter"></a>
                <a href="#" class="fab fa-instagram"></a>
                <a href="#" class="fab fa-linkedin"></a>
            </div>
        </div>

        <div class="box">
            <img src="image/doc-3.jpg" alt="">
            <h3>john deo</h3>
```

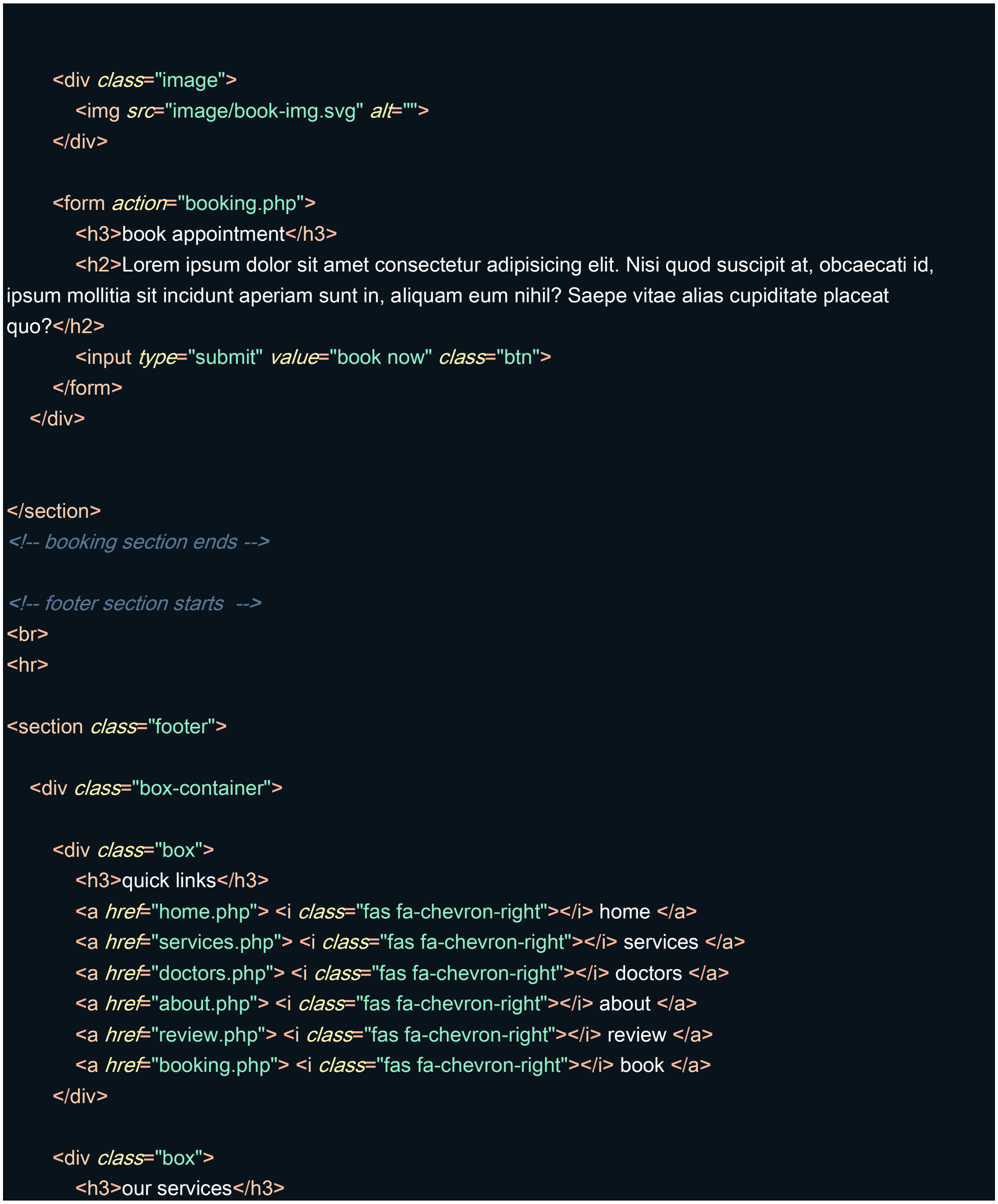
```
      <span>expert doctor</span>
      <div class="share">
        <a href="#" class="fab fa-facebook-f"></a>
        <a href="#" class="fab fa-twitter"></a>
        <a href="#" class="fab fa-instagram"></a>
        <a href="#" class="fab fa-linkedin"></a>
      </div>
    </div>
```



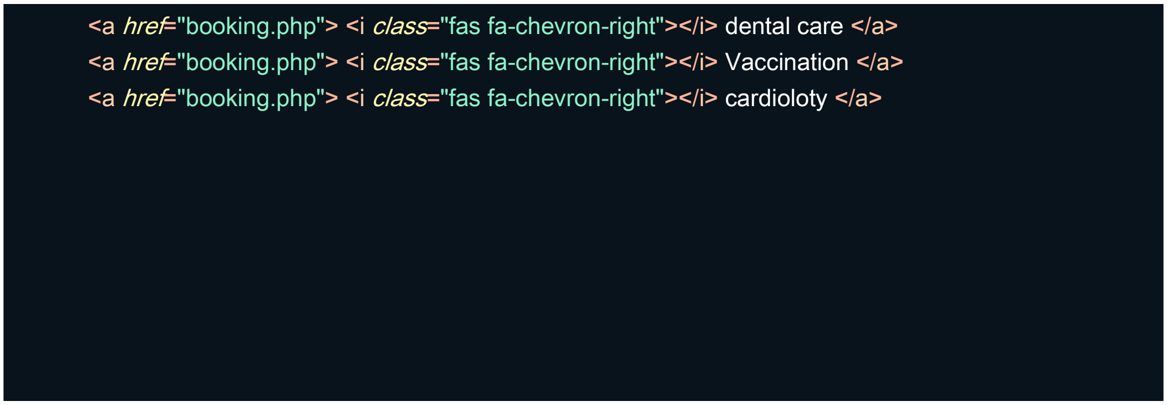
```
    <div class="box">
      <img src="image/doc-4.jpg" alt="">
      <h3>john deo</h3>
      <span>expert doctor</span>
      <div class="share">
        <a href="#" class="fab fa-facebook-f"></a>
        <a href="#" class="fab fa-twitter"></a>
        <a href="#" class="fab fa-instagram"></a>
        <a href="#" class="fab fa-linkedin"></a>
      </div>
    </div>

    <div class="box">
      <img src="image/doc-5.jpg" alt="">
      <h3>john deo</h3>
      <span>expert doctor</span>
      <div class="share">
        <a href="#" class="fab fa-facebook-f"></a>
        <a href="#" class="fab fa-twitter"></a>
        <a href="#" class="fab fa-instagram"></a>
        <a href="#" class="fab fa-linkedin"></a>
      </div>
    </div>

    <div class="box">
      <img src="image/doc-6.jpg" alt="">
      <h3>john deo</h3>
      <span>expert doctor</span>
      <div class="share">
        <a href="#" class="fab fa-facebook-f"></a>
        <a href="#" class="fab fa-twitter"></a>
        <a href="#" class="fab fa-instagram"></a>
        <a href="#" class="fab fa-linkedin"></a>
      </div>
    </div>
  </div>
</section>
```



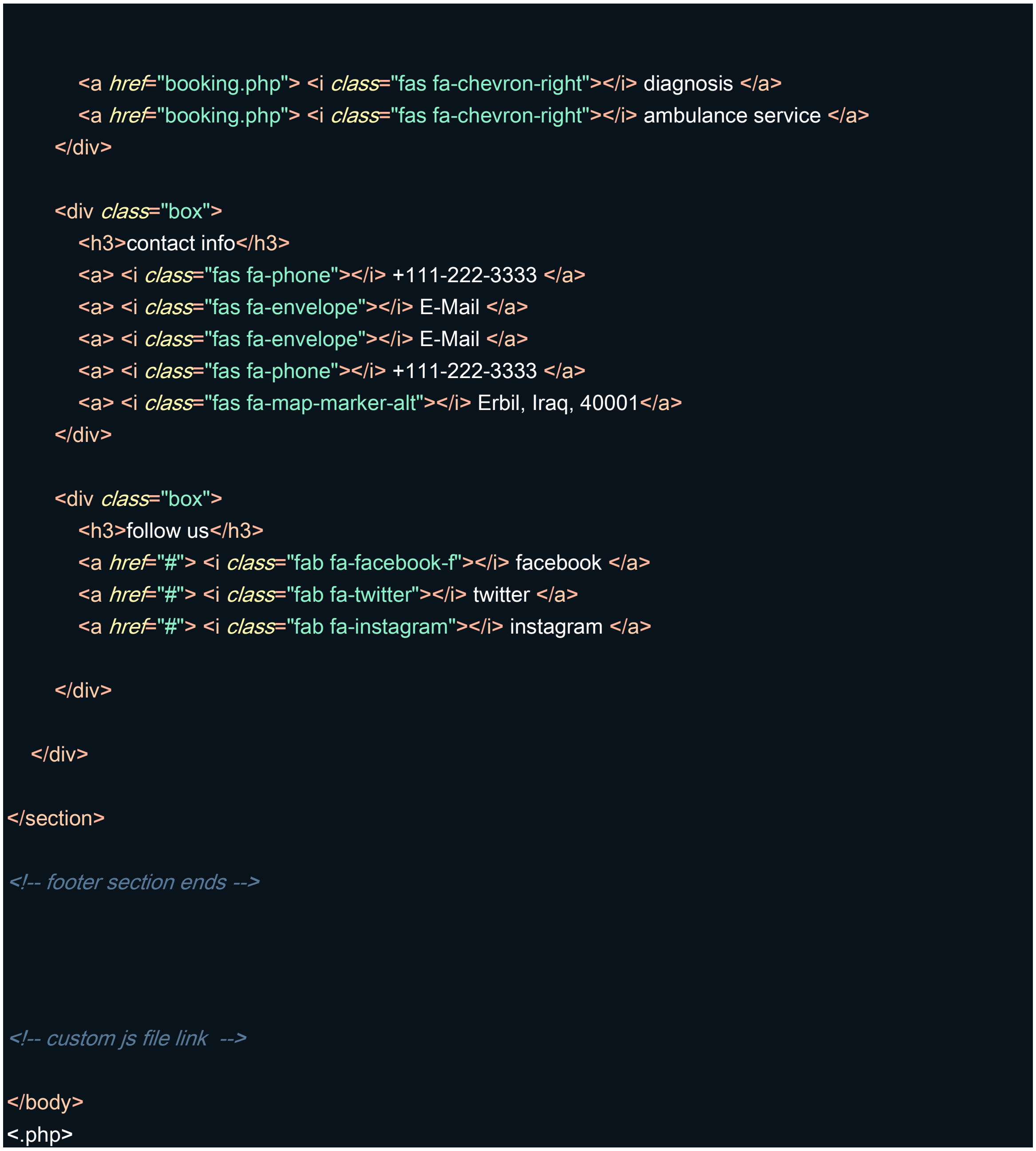
```
<!-- doctors section ends -->

<!-- booking section starts   -->

<section class="book" id="book">

    <h1 class="heading"> <span>Book</span> Appoinment </h1>

    <div class="row">
```



```


        <div class="image">
            <img src="image/book-img.svg" alt="">
        </div>

        <form action="booking.php">
            <h3>book appointment</h3>
            <h2>Lorem ipsum dolor sit amet consectetur adipisicing elit. Nisi quod suscipit at, obcaecati id,
ipsum mollitia sit incidunt aperiam sunt in, aliquam eum nihil? Saepe vitae alias cupiditate placeat
quo?</h2>
            <input type="submit" value="book now" class="btn">
        </form>
    </div>


</section>
<!-- booking section ends -->

<!-- footer section starts  -->
<br>
<hr>

<section class="footer">

    <div class="box-container">

        <div class="box">
            <h3>quick links</h3>
            <a href="home.php"> <i class="fas fa-chevron-right"></i> home </a>
            <a href="services.php"> <i class="fas fa-chevron-right"></i> services </a>
            <a href="doctors.php"> <i class="fas fa-chevron-right"></i> doctors </a>
            <a href="about.php"> <i class="fas fa-chevron-right"></i> about </a>
            <a href="review.php"> <i class="fas fa-chevron-right"></i> review </a>
            <a href="booking.php"> <i class="fas fa-chevron-right"></i> book </a>
        </div>

        <div class="box">
            <h3>our services</h3>
```

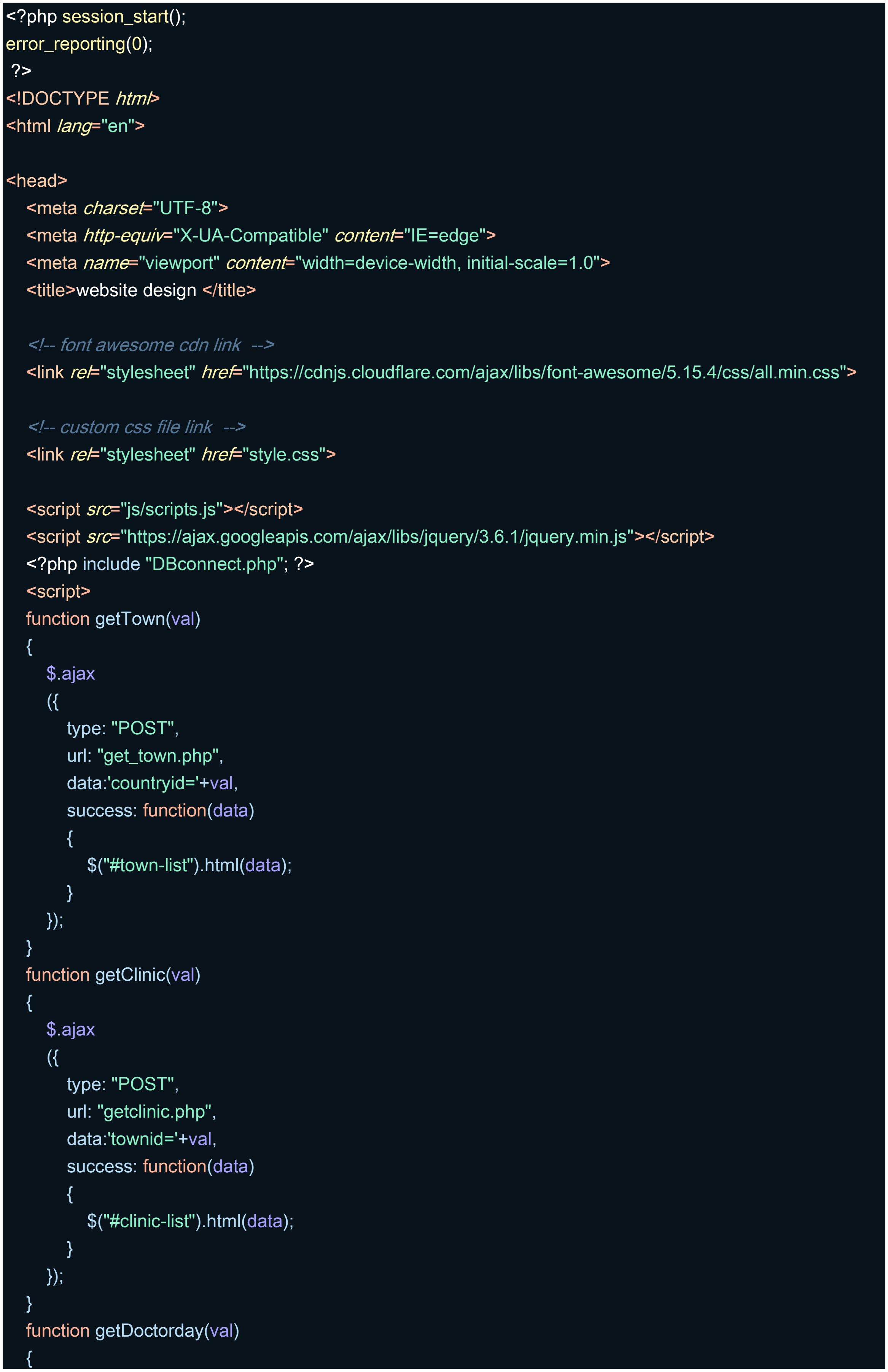


```
      <a href="booking.php"> <i class="fas fa-chevron-right"></i> dental care </a>
      <a href="booking.php"> <i class="fas fa-chevron-right"></i> Vaccination </a>
      <a href="booking.php"> <i class="fas fa-chevron-right"></i> cardioloty </a>
```

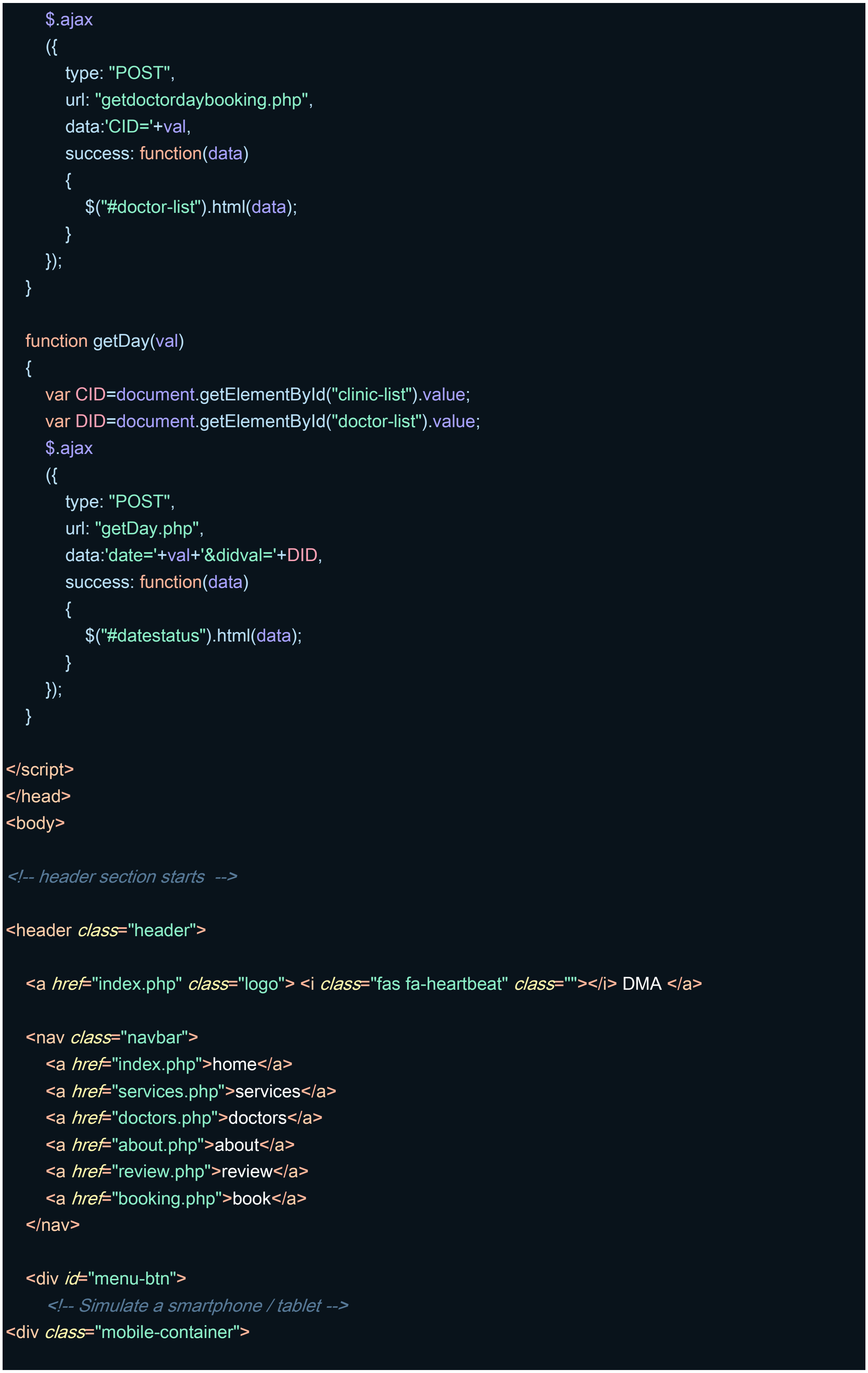


```
      <a href="booking.php"> <i class="fas fa-chevron-right"></i> diagnosis </a>
      <a href="booking.php"> <i class="fas fa-chevron-right"></i> ambulance service </a>
    </div>

    <div class="box">
      <h3>contact info</h3>
      <a> <i class="fas fa-phone"></i> +111-222-3333 </a>
      <a> <i class="fas fa-envelope"></i> E-Mail </a>
      <a> <i class="fas fa-envelope"></i> E-Mail </a>
      <a> <i class="fas fa-phone"></i> +111-222-3333 </a>
      <a> <i class="fas fa-map-marker-alt"></i> Erbil, Iraq, 40001</a>
    </div>

    <div class="box">
      <h3>follow us</h3>
      <a href="#"> <i class="fab fa-facebook-f"></i> facebook </a>
      <a href="#"> <i class="fab fa-twitter"></i> twitter </a>
      <a href="#"> <i class="fab fa-instagram"></i> instagram </a>

    </div>

  </div>

</section>

<!-- footer section ends -->




<!-- custom js file link  -->

</body>
<.php>
```

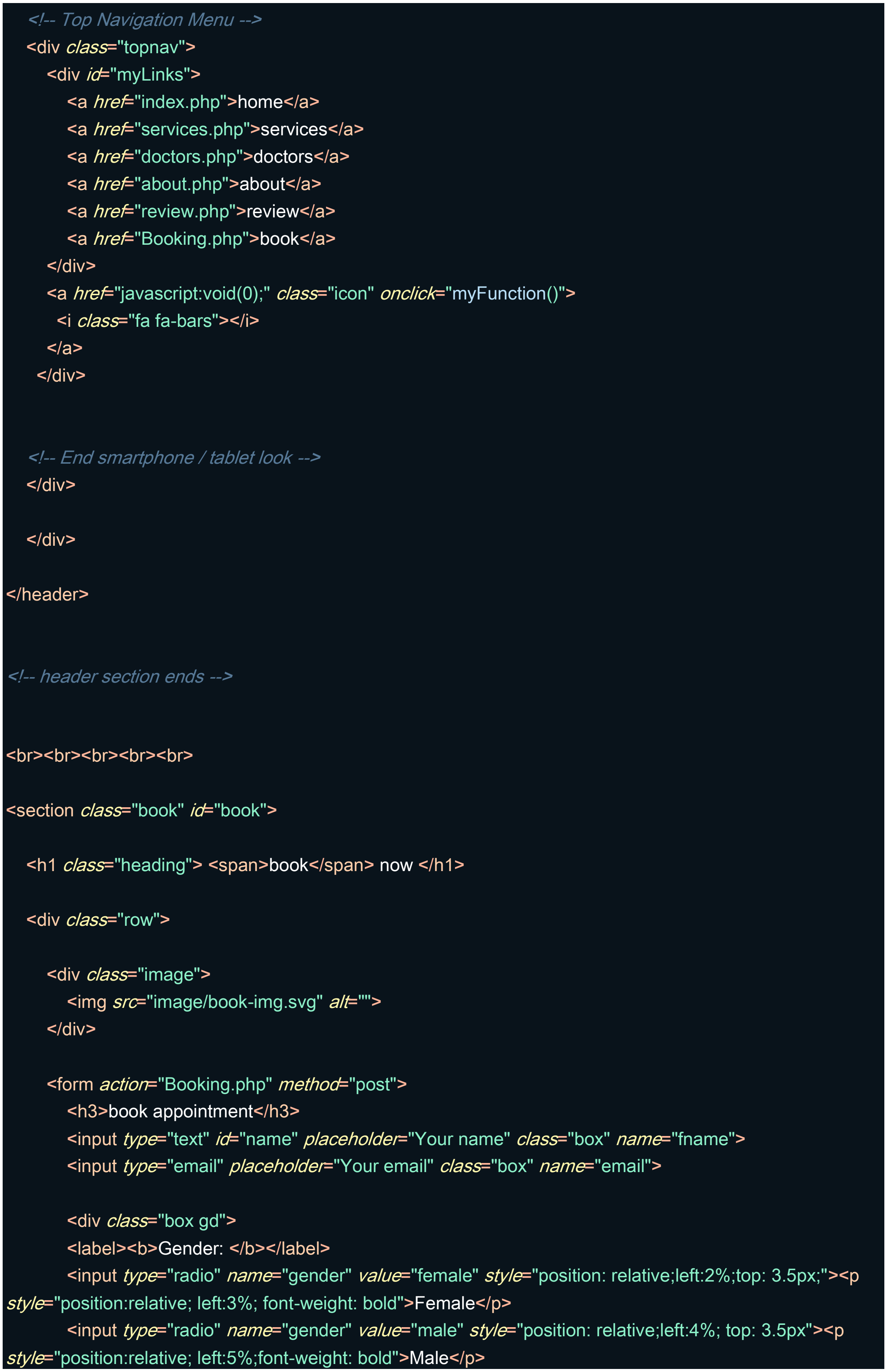
Booking.php: Figure 3.6



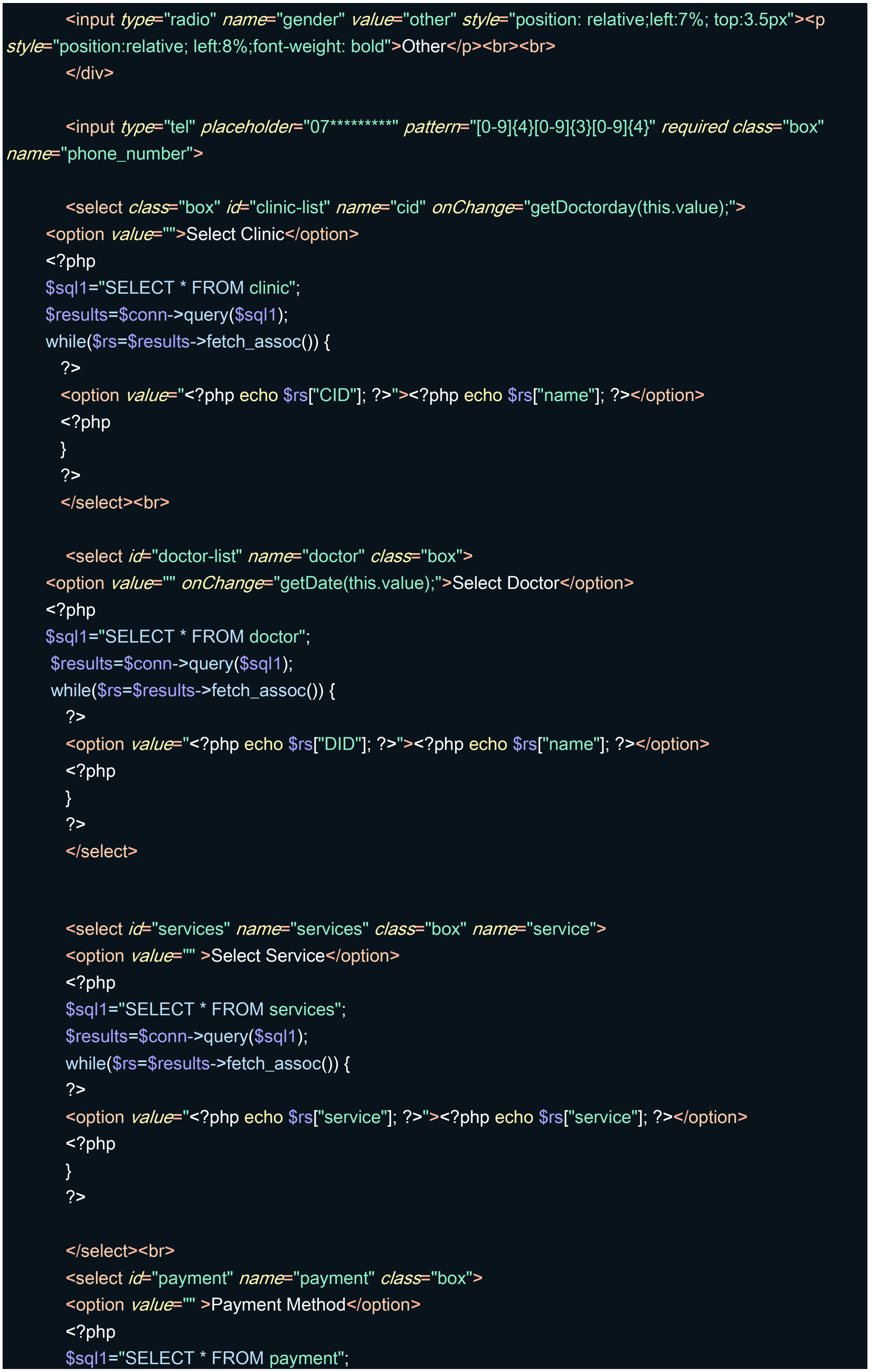
```
<?php session_start();
error_reporting(0);
 ?>
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>website design </title>

    <!-- font awesome cdn link  -->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css">

    <!-- custom css file link  -->
    <link rel="stylesheet" href="style.css">

    <script src="js/scripts.js"></script>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jquery.min.js"></script>
    <?php include "DBconnect.php"; ?>
    <script>
    function getTown(val)
    {
        $.ajax
        ({
            type: "POST",
            url: "get_town.php",
            data:'countryid='+val,
            success: function(data)
            {
                $("#town-list").html(data);
            }
        });
    }
    function getClinic(val)
    {
        $.ajax
        ({
            type: "POST",
            url: "getclinic.php",
            data:'townid='+val,
            success: function(data)
            {
                $("#clinic-list").html(data);
            }
        });
    }
    function getDoctorday(val)
    {
```



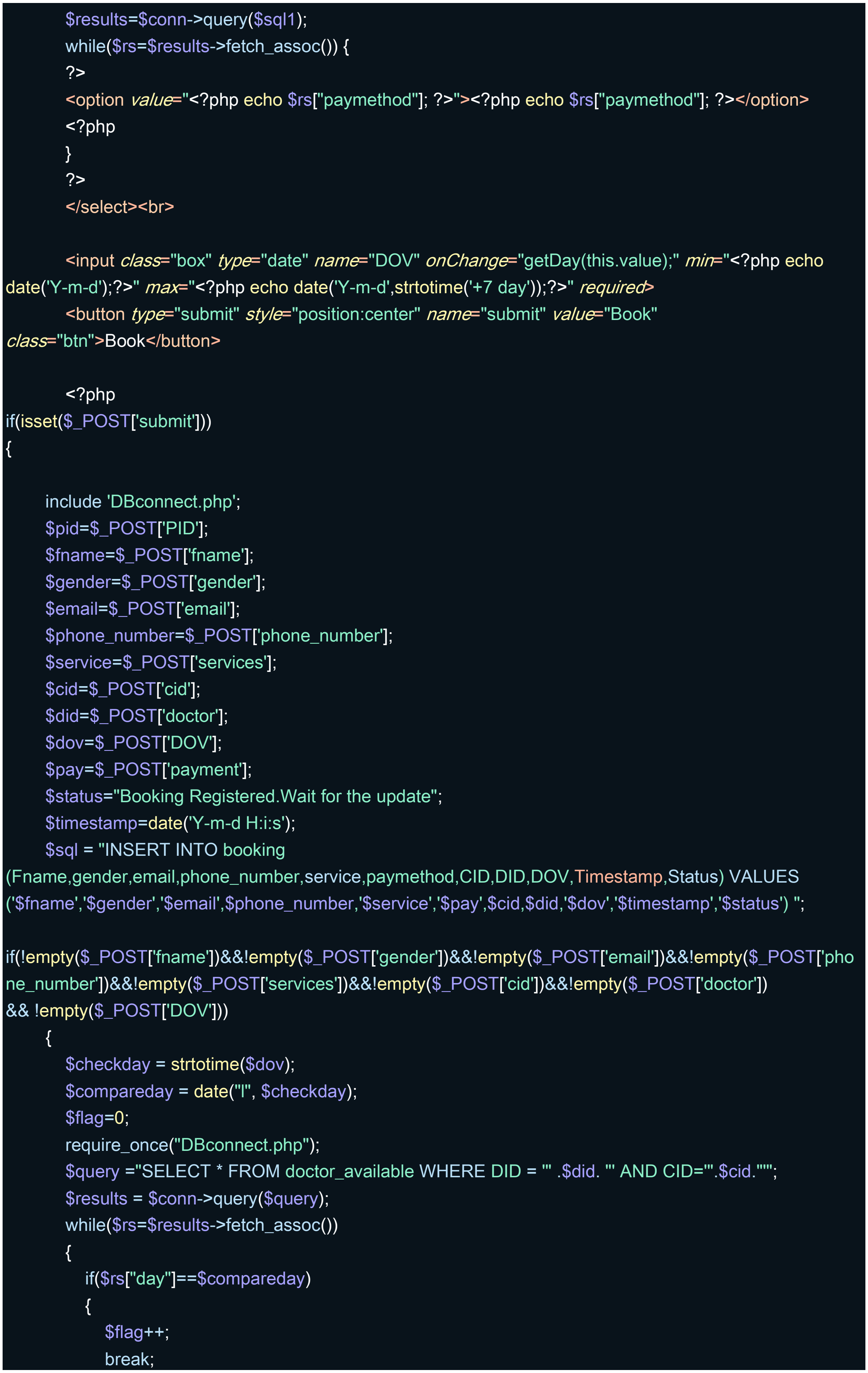
```
        $.ajax
        ({
            type: "POST",
            url: "getdoctordaybooking.php",
            data:'CID='+val,
            success: function(data)
            {
                $("#doctor-list").html(data);
            }
        });
    }

    function getDay(val)
    {
        var CID=document.getElementById("clinic-list").value;
        var DID=document.getElementById("doctor-list").value;
        $.ajax
        ({
            type: "POST",
            url: "getDay.php",
            data:'date='+val+'&didval='+DID,
            success: function(data)
            {
                $("#datestatus").html(data);
            }
        });
    }

</script>
</head>
<body>

<!-- header section starts  -->

<header class="header">

    <a href="index.php" class="logo"> <i class="fas fa-heartbeat" class=""></i> DMA </a>

    <nav class="navbar">
        <a href="index.php">home</a>
        <a href="services.php">services</a>
        <a href="doctors.php">doctors</a>
        <a href="about.php">about</a>
        <a href="review.php">review</a>
        <a href="booking.php">book</a>
    </nav>

    <div id="menu-btn">
        <!-- Simulate a smartphone / tablet -->
<div class="mobile-container">
```



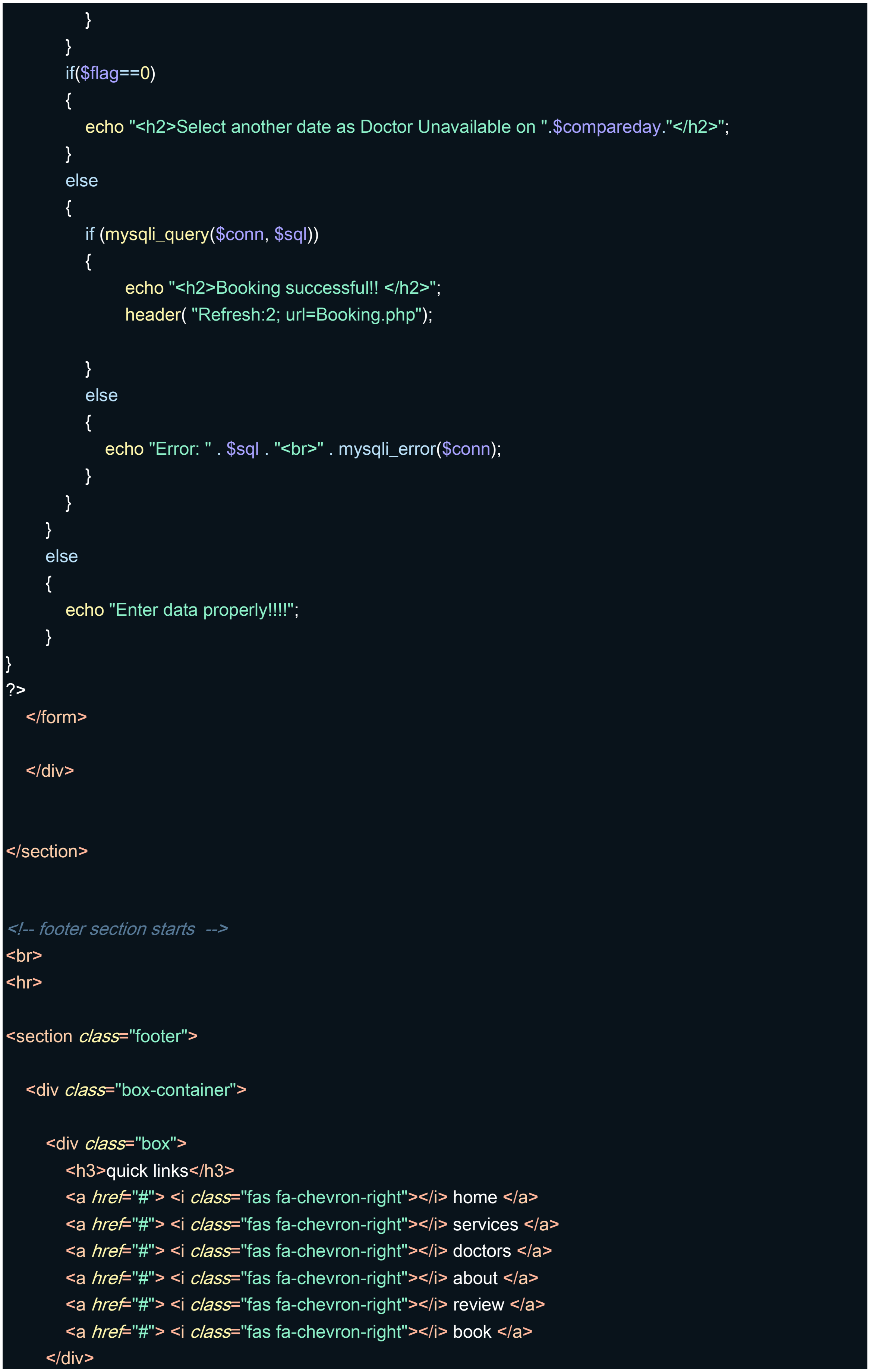
```
    <!-- Top Navigation Menu -->
    <div class="topnav">
        <div id="myLinks">
            <a href="index.php">home</a>
            <a href="services.php">services</a>
            <a href="doctors.php">doctors</a>
            <a href="about.php">about</a>
            <a href="review.php">review</a>
            <a href="Booking.php">book</a>
        </div>
        <a href="javascript:void(0);" class="icon" onclick="myFunction()">
          <i class="fa fa-bars"></i>
        </a>
      </div>


    <!-- End smartphone / tablet look -->
    </div>

    </div>

</header>


<!-- header section ends -->


<br><br><br><br><br>

<section class="book" id="book">

    <h1 class="heading"> <span>book</span> now </h1>

    <div class="row">

        <div class="image">
            <img src="image/book-img.svg" alt="">
        </div>

        <form action="Booking.php" method="post">
            <h3>book appointment</h3>
            <input type="text" id="name" placeholder="Your name" class="box" name="fname">
            <input type="email" placeholder="Your email" class="box" name="email">

            <div class="box gd">
            <label><b>Gender: </b></label>
            <input type="radio" name="gender" value="female" style="position: relative;left:2%;top: 3.5px;"><p
style="position:relative; left:3%; font-weight: bold">Female</p>
            <input type="radio" name="gender" value="male" style="position: relative;left:4%; top: 3.5px"><p
style="position:relative; left:5%;font-weight: bold">Male</p>
```



```
          <input type="radio" name="gender" value="other" style="position: relative;left:7%; top:3.5px"><p
style="position:relative; left:8%;font-weight: bold">Other</p><br><br>
          </div>

          <input type="tel" placeholder="07*********" pattern="[0-9]{4}[0-9]{3}[0-9]{4}" required class="box"
name="phone_number">

          <select class="box" id="clinic-list" name="cid" onChange="getDoctorday(this.value);">
       <option value="">Select Clinic</option>
       <?php
       $sql1="SELECT * FROM clinic";
       $results=$conn->query($sql1);
       while($rs=$results->fetch_assoc()) {
         ?>
         <option value="<?php echo $rs["CID"]; ?>"><?php echo $rs["name"]; ?></option>
         <?php
         }
         ?>
         </select><br>

          <select id="doctor-list" name="doctor" class="box">
       <option value="" onChange="getDate(this.value);">Select Doctor</option>
       <?php
       $sql1="SELECT * FROM doctor";
        $results=$conn->query($sql1);
        while($rs=$results->fetch_assoc()) {
          ?>
          <option value="<?php echo $rs["DID"]; ?>"><?php echo $rs["name"]; ?></option>
          <?php
          }
          ?>
          </select>


          <select id="services" name="services" class="box" name="service">
          <option value="" >Select Service</option>
          <?php
          $sql1="SELECT * FROM services";
          $results=$conn->query($sql1);
          while($rs=$results->fetch_assoc()) {
          ?>
          <option value="<?php echo $rs["service"]; ?>"><?php echo $rs["service"]; ?></option>
          <?php
          }
          ?>

          </select><br>
          <select id="payment" name="payment" class="box">
          <option value="" >Payment Method</option>
          <?php
          $sql1="SELECT * FROM payment";
```



```
            $results=$conn->query($sql1);
            while($rs=$results->fetch_assoc()) {
            ?>
            <option value="<?php echo $rs["paymethod"]; ?>"><?php echo $rs["paymethod"]; ?></option>
            <?php
            }
            ?>
            </select><br>

            <input class="box" type="date" name="DOV" onChange="getDay(this.value);" min="<?php echo
date('Y-m-d');?>" max="<?php echo date('Y-m-d',strtotime('+7 day'));?>" required>
            <button type="submit" style="position:center" name="submit" value="Book"
class="btn">Book</button>

            <?php
if(isset($_POST['submit']))
{

        include 'DBconnect.php';
        $pid=$_POST['PID'];
        $fname=$_POST['fname'];
        $gender=$_POST['gender'];
        $email=$_POST['email'];
        $phone_number=$_POST['phone_number'];
        $service=$_POST['services'];
        $cid=$_POST['cid'];
        $did=$_POST['doctor'];
        $dov=$_POST['DOV'];
        $pay=$_POST['payment'];
        $status="Booking Registered.Wait for the update";
        $timestamp=date('Y-m-d H:i:s');
        $sql = "INSERT INTO booking
(Fname,gender,email,phone_number,service,paymethod,CID,DID,DOV,Timestamp,Status) VALUES
('$fname','$gender','$email',$phone_number,'$service','$pay',$cid,$did,'$dov','$timestamp','$status') ";

if(!empty($_POST['fname'])&&!empty($_POST['gender'])&&!empty($_POST['email'])&&!empty($_POST['pho
ne_number'])&&!empty($_POST['services'])&&!empty($_POST['cid'])&&!empty($_POST['doctor'])
&& !empty($_POST['DOV']))
        {
            $checkday = strtotime($dov);
            $compareday = date("l", $checkday);
            $flag=0;
            require_once("DBconnect.php");
            $query ="SELECT * FROM doctor_available WHERE DID = '" .$did. "' AND CID='".$cid."'";
            $results = $conn->query($query);
            while($rs=$results->fetch_assoc())
            {
                if($rs["day"]==$compareday)
                {
                    $flag++;
                    break;
```



```
                }
            }
            if($flag==0)
            {
                echo "<h2>Select another date as Doctor Unavailable on ".$compareday."</h2>";
            }
            else
            {
                if (mysqli_query($conn, $sql))
                {
                        echo "<h2>Booking successful!! </h2>";
                        header( "Refresh:2; url=Booking.php");

                }
                else
                {
                    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
                }
            }
        }
        else
        {
            echo "Enter data properly!!!!";
        }
}
?>
    </form>

    </div>


</section>


<!-- footer section starts  -->
<br>
<hr>

<section class="footer">

    <div class="box-container">

        <div class="box">
            <h3>quick links</h3>
            <a href="#"> <i class="fas fa-chevron-right"></i> home </a>
            <a href="#"> <i class="fas fa-chevron-right"></i> services </a>
            <a href="#"> <i class="fas fa-chevron-right"></i> doctors </a>
            <a href="#"> <i class="fas fa-chevron-right"></i> about </a>
            <a href="#"> <i class="fas fa-chevron-right"></i> review </a>
            <a href="#"> <i class="fas fa-chevron-right"></i> book </a>
        </div>
        <div class="box">
            <h3>our services</h3>
            <a href="#"> <i class="fas fa-chevron-right"></i> dental care </a>
            <a href="#"> <i class="fas fa-chevron-right"></i> Vaccination </a>
            <a href="#"> <i class="fas fa-chevron-right"></i> cardioloty </a>
            <a href="#"> <i class="fas fa-chevron-right"></i> diagnosis </a>
            <a href="#"> <i class="fas fa-chevron-right"></i> ambulance service </a>
        </div>

        <div class="box">
            <h3>contact info</h3>
            <a href="#"> <i class="fas fa-phone"></i> +111-222-3333 </a>
            <a href="#"> <i class="fas fa-phone"></i> +111-222-3333 </a>
            <a href="#"> <i class="fas fa-envelope"></i> E-Mail </a>
            <a href="#"> <i class="fas fa-envelope"></i> E-Mail </a>
            <a href="#"> <i class="fas fa-map-marker-alt"></i> Erbil, Iraq, 40001</a>
        </div>

        <div class="box">
            <h3>follow us</h3>
            <a href="#"> <i class="fab fa-facebook-f"></i> facebook </a>
            <a href="#"> <i class="fab fa-twitter"></i> twitter </a>
            <a href="#"> <i class="fab fa-instagram"></i> instagram </a>

        </div>

    </div>

</section>

<!-- footer section ends -->
<br><br><br><br/br>
</body>
</html>
```

DBconnect.php: (for connection establishment with database)

```
<?php
$conn = mysqli_connect('localhost','root','','appointment');

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>
```