Higher Nationals

Internal verification of assessment decisions – BTEC (RQF)

INTERNAL VERIFICATION – ASSESSMENT DECISIONS					
Programme title	BTEC Higher National Diploma in Computing				
Assessor	Mr. Jeykanth		**Internal Verifier**	Mr.Lakindu Premachandra	
Unit(s)	Unit 10: Web Design and Development				
Assignment title	Online Hospital Management System				
Student'sname	Ranudi Gayathmie Kariyapperuma				
List which assessment criteria the Assessor has awarded.	**Pass**		**Merit**		**Distinction**
INTERNAL VERIFIER CHECKLIST					
Do the assessment criteria awarded match those shown in the assignment brief?	Y/N				
Is the Pass/Merit/Distinction grade awarded justified by the assessor's comments on the student work?	Y/N				
Has the work been assessed accurately?	Y/N				
Is the feedback to the student: Give details: • Constructive? • Linked to relevant assessment criteria? • Identifying opportunities for improved performance? • Agreeing actions?	Y/N Y/N Y/N Y/N				
Does the assessment decision need amending?	Y/N				
Assessor signature			**Date**		
Internal Verifier signature			**Date**		
Programme Leader signature(if required)			**Date**		

Confirm action completed			
Remedial action taken Give details:			
Assessor signature		**Date**	
Internal Verifier signature		**Date**	
Programme Leader signature (if required)		**Date**	

Higher Nationals - Summative Assignment Feedback Form

Student Name/ID	Ranudi Gayathmie Kariyapperuma KIR/X -00104243		
Unit Title	Unit 10: Website Design & Development		
Assignment Number	1	**Assessor**	
Submission Date	31.12.2023	**Date Received 1st submission**	
Re-submission Date		**Date Received 2nd submission**	

Assessor Feedback:

LO1. Explain server technologies and management services associated with hosting and managing websites.

Pass, Merit & Distinction Descripts	P1 ☐	P2 ☐	M1 ☐	M2 ☐	D1 ☐

LO2. Categorise website technologies, tools and software used to develop websites.

Pass, Merit & Distinction Descripts	P3 ☐	P4 ☐	M3 ☐	D1 ☐

LO3. Utilise website technologies, tools and techniques with good design principles to create a multipage website.

Pass, Merit & Distinction Descripts	P5 ☐	P6 ☐	M4 ☐	D2 ☐

LO4. Create and use a Test Plan to review the performance and design of a multipage website.

Pass, Merit & Distinction Descripts	P7 ☐	M5 ☐	D3 ☐

Grade:	**Assessor Signature:**	**Date:**
Resubmission Feedback:		
Grade:	**Assessor Signature:**	**Date:**
Internal Verifier's Comments:		
Signature & Date:		

* Please note that grade decisions are provisional. They are only confirmed once internal and external moderation has taken place and grades decisions have been agreed at the assessment board

Assignment Feedback

Formative Feedback: Assessor to Student			
Action Plan			
Summative feedback			
Feedback: Student to Assessor			
Assessor signature		**Date**	
Student signature		**Date**	

Pearson
Higher Nationals in
Computing

Unit 10: Web Design and Development
Assignment 01

General Guidelines

1. A Cover page or title page – You should always attach a title page to your assignment. Use previous page as your cover sheet and make sure all the details are accurately filled.
2. Attach this brief as the first section of your assignment.
3. All the assignments should be prepared using a word processing software.
4. All the assignments should be printed on A4 sized papers. Use single side printing.
5. Allow 1" for top, bottom , right margins and 1.25" for the left margin of each page.

Word Processing Rules

1. The font size should be **12** point, and should be in the style of **Time New Roman**.
2. **Use 1.5 line spacing**. Left justify all paragraphs.
3. Ensure that all the headings are consistent in terms of the font size and font style.
4. Use **footer function in the word processor to insert Your Name, Subject, Assignment No, and Page Number on each pag**e. This is useful if individual sheets become detached for any reason.
5. Use word processing application spell check and grammar check function to help editing your assignment.

Important Points:

1. It is strictly prohibited to use textboxes to add texts in the assignments, except for the compulsory information. eg: Figures, tables of comparison etc. Adding text boxes in the body except for the before mentioned compulsory information will result in rejection of your work.
2. Carefully check the hand in date and the instructions given in the assignment. Late submissions will not be accepted.
3. Ensure that you give yourself enough time to complete the assignment by the due date.
4. Excuses of any nature will not be accepted for failure to hand in the work on time.
5. You must take responsibility for managing your own time effectively.
6. If you are unable to hand in your assignment on time and have valid reasons such as illness, you may apply (in writing) for an extension.
7. Failure to achieve at least PASS criteria will result in a REFERRAL grade .
8. Non-submission of work without valid reasons will lead to an automatic RE FERRAL. You will then be asked to complete an alternative assignment.
9. If you use other people's work or ideas in your assignment, reference them properly using HARVARD referencing system to avoid plagiarism. You have to provide both in-text citation and a reference list.
10. If you are proven to be guilty of plagiarism or any academic misconduct, your grade could be reduced to A REFERRAL or at worst you could be expelled from the course.
11. If you are proven to be guilty of plagiarism or any academic misconduct, your grade could be reduced to A REFERRAL or at worst you could be expelled from the course.

Student Declaration

I hereby, declare that I know what plagiarism entails, namely to use another's work and to present it as my own without attributing the sources in the correct way. I further understand what it means to copy another's work.

1. I know that plagiarism is a punishable offence because it constitutes theft.
2. I understand the plagiarism and copying policy of the Pearson UK.
3. I know what the consequences will be if I plagiaries or copy another's work in any of the assignments for this program.
4. I declare therefore that all work presented by me for every aspects of my program, will be my own, and where I have made use of another's work, I will attribute the source in the correct way.
5. I acknowledge that the attachment of this document signed or not, constitutes a binding agreement between myself and Pearson, UK.
6. I understand that my assignment will not be considered as submitted if this document is not attached to the attached.

ranudigk@gmail.com
Student's Signature: **Date:** 31.12.2023
(*Provide E-mail ID*) **(*Provide Submission Date*)**

Student Name /ID Number	Ranudi Gayathmie Kariyapperuma KIR/X -00104243
Unit Number and Title	Unit 10- Web Design and Development
Academic Year	2021/2022
Unit Tutor	Mr. Jeykanth
Assignment Title	Online Hospital Management System
Issue Date	05.11.2023
Submission Date	31.12.2023
IV Name & Date	
Submission Format:	
Part 1. Report- Submit a professional report with approepriate report formattimg and guidelines followed. All the research data should be referenced along with in-text citations using Hrvard referencing syste. Part 2 A fully functional web solution	

Assignment Brief

Scenario.

'Apex Design Works' is a leading web design and marketing company. They are focusing on helping businesses communicate more effectively and build their business through a creative design. Assume that, you work as an apprentice web developer for Apex Web Design and marketing company. As part of your role, you have been asked to create a Website for the following organization .

Arogya Health Care hospital currently uses a manual system for the management and maintenance of critical information. The current system requires numerous paper forms, with data stores spread throughout the hospital management infrastructure. Often information (on forms) is incomplete, or does not follow management standards. Multiple copies of the same information exist in the hospital and may lead to inconsistencies in data in various data stores. There are number of documents to be maintained in the Health Care hospital and this information typically involves; patient personal information and medical history, staff information, room and ward scheduling, staff scheduling, operating theater scheduling and various facilities waiting lists. All of this information must be managed in an efficient and cost wise fashion so that the resources can be managed effectively. The reception module handles various inquiries about the patient's admission and discharge details, and the patient's movements within the hospital.
Assume that you are the web developer hired by the Health Care Hospital, to propose, and engineer a low cost but yet powerful and complete Hospital Management System (HMS) for the scenario given above. Suggest and implement important functionalities and features to the system by identifying system functionalities.

The new system is to control the following information

- patient information
- room availability
- staff and operating room schedules
- patient invoices

Develop a web based solution for the above scenario and produce a report covering the following tasks.

Task 1 - Server technologies and management services associated with hosting and managing websites (LO1)

1.1 Explain and differentiate the different web technologies such as communication protocols, server hardware, operating systems and web server software with regards to designing, publishing and accessing the Hospital Management System (HMS).

1.2 Identifry and define the types of DNS and the uses of it, with clarifications on how domain names are structured. Review the effect of search engines on website performance. Provide evidence-based support for improving a site's index value and rank of the Hospital Management System (HMS) through search engine optimization.

1.3 Identify and explain the common web development technologies and frameworks. Explain the tools and techniques chosen to develop the above web application and justify your choice by providing valid evidences

Task 2 - Categories website technologies, tools and software used to develop websites (LO2)

2.1 Considering the requirments given in the above scenario define the relationships between front-end and back-end website technologies and discuss how the front-end and the back-end relate to presentation and application layers.

2.2 Discuss the differences between online website creation tools and custom-built web sites by considering the design flexibility, performance, functionality, User Experience (UX) and User Interface (UI). Evaluate the tools and techniques available to design the web application gicven in the scenario.

.

Task 3 - Utilize website technologies, tools and techniques with good design principles to create a multipage website (LO3)

3.1 Design a suitable web application solution for the given scenario using PHP, JS and MySQL (Screenshots of important code lines with proper comments and user interfaces filled with sample data must be attached to the documentation). Apply a database design for the proposed system and provide the well normalized database design of the proposed system. Provide evidences of the design, multipage website supported with fidelity wireframes and a full set of client and user requirements.

3.2 Compare and contrast the multipage website created to the design document. Use your design document with appropriate principles, standards and guidelines to produce a branded, multipage website supported with realistic content and Critically evaluate the web design ,development process against your design document analyisng any technical challenges you faced during the development.

Note - Synthesize client and the server-side functionalities in the proposed design.

Task 4 - Create and use a Test Plan to review the performance and design of a multipage website (LO4)

4.1 QA process is expected to discover design issues and development errors while testing a product's user interface (UI) and gauging the user experience (UX). Evaluate the Quality Assurance (QA) process and review how it was implemented during your design and development stages.

4.2 Create a suitable test plan for the developed system and critically evaluate the results of your Test Plan . Include a review of the overall success of your multipage website; use this evaluation to explain any areas of success and provide justified recommendations for areas that require improvements.

Observation Sheet

Activity No	Activity	Learning Outcome	Feedback (Pass/ Redo)
1	Explain server technologies and management services associated with hosting and managing websites.	LO1	
2	categorize website technologies, tools and software used to develop websites.	LO2	
3	Utilize website technologies, tools and techniques with good design principles to create a multipage website.	LO3	
4	Create and use a Test Plan to review the performance and design of a multipage website.	LO4	

Comments:

Assessor Name :…….
Date :…….
Assessor Signature :…….

ACKKNOWLEDGEMENT

At last author would like to share the experience while doing the project. Author learns many new things about the networking topics. The best thing which author can share is that author developed more interest in this subject. This Project gave author experience of how to do an event . A very special thanks to Mr. Jeykanth who teach us this subject and Author thanks for who helped author to do this kind of project. Thank you!

Regards,
The author,
Ranudi Kariyapperuma

Table of Contents

Tables of Figures

List of Tables

The Introduction for DNS

According to the internet's phone book, the Domain Name System (DNS) converts easily navigable domain names, such as google.com, into the numerical IP addresses, such as 192.168.1.1, that are used by devices for communication. Your device contacts a DNS resolver to determine the IP address linked to a domain name that you enter into your browser. Prior to consulting a hierarchy of servers, the resolver first clears its cache. Root nameservers point to TLD nameservers (such as.com), which in turn point to particular domain nameservers (such as google.com). Your device may connect to the desired website thanks to these servers, which save the IP addresses associated with the requested domain. By converting memorable domain names into IP addresses that computers use to find one another on the internet, DNS makes navigation smooth and easy.

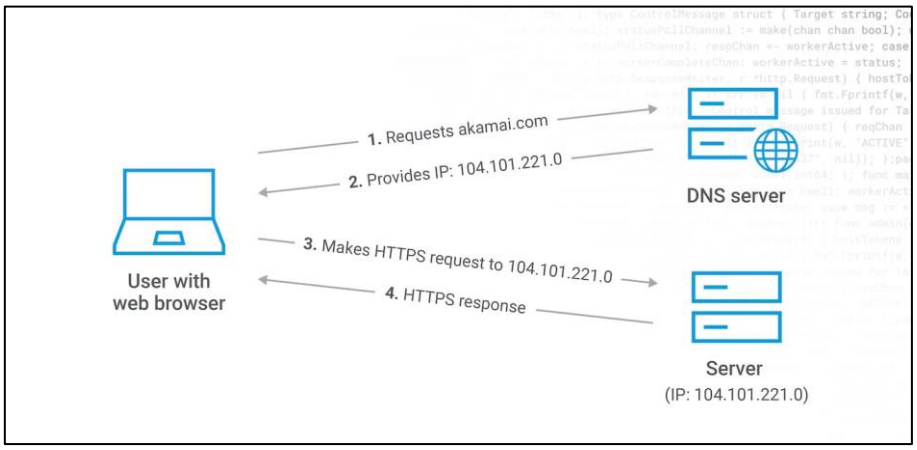


Figure 1 : DNS

The way DNS Works

A DNS resolver (often offered by the user's internet service provider or a third-party like Google's 8.8.8.8) is contacted by the user's computer when the user types a domain name into their browser. The resolver starts the procedure by looking up the relevant IP address in its cache. It turns into a little bit of an investigator if the information is missing.

Recursive Resolvers

These resolvers function similarly to the DNS industry's investigators. The recursive resolver is the first resource a user sees when they enter a domain name. It starts by looking

up the relevant IP address in its cache. It sets out to locate the IP if it isn't cached. In order to obtain the required data, it first makes contact with other DNS servers and navigates the DNS hierarchy.

Root Nameservers

In case it lacks the required information, the recursive resolver gets in touch with a root nameserver. In the DNS hierarchy, these servers are at the top. Globally, there are just 13 sets of root nameservers. They provide details regarding top-level domains (TLDs), such as.com,.org,.net, and so on. Rather than having unique IP addresses, they offer details on where each TLD's authoritative nameservers may be found.

TLD Nameservers

The resolver makes contact with the TLD nameservers after being directed by the root nameserver to the authoritative nameservers for a certain TLD. These servers have comprehensive data regarding domains under that TLD. For example, the TLD nameserver for.com will point the resolver to the authoritative nameserver for google.com if you search for google.com.

Authoritative Nameservers

The resolver contacts the authoritative nameserver for the particular domain (such as google.com) being queried as the last stage in the DNS resolution process. The precise IP address linked to the domain that was requested is stored on this nameserver. When the needed IP address is obtained from the authoritative nameserver, it follows the same route back to the recursive resolver and, eventually, the user's computer. By connecting the browser to the desired website with this IP address, the user can view the content.

By converting domain names into the matching IP addresses needed for internet connection, each of these parts works together in an organized way to support DNS's smooth operation and ensure that users may visit websites with ease.

DNS server Types

- **Recursive Resolvers**

In the DNS resolution process, recursive resolvers are essential. They provide as a bridge between the DNS infrastructure and consumers' devices, including PCs, phones, and routers. Recursive resolvers are the first to be consulted when a user enters a domain name into a browser to commence a request. To find out if it already has the IP address for the requested domain, it first looks in its cache. It starts the DNS resolution procedure if not. In order to learn about the TLD nameservers, these resolvers are made to follow the DNS hierarchy, which begins with querying root nameservers. After that, they move down the hierarchy until they get to the nameservers that are in charge of that particular domain.To provide an IP address that corresponds to a domain name to a user, recursive resolvers must compile and organize the data that they have acquired from various DNS server tiers. They are essential for quickly and effectively answering questions and providing easy access to websites and online services.

Usually found in ISP networks, these resolvers are also offered by outside providers like Cloudflare DNS and Google Public DNS. In order to effectively manage and cache DNS data, they use complex algorithms that shorten query times and improve query resolution. Recursive resolvers also employ security features such as DNSSEC (Domain Name System Security Extensions) to guarantee the veracity and accuracy of the resolved DNS data, hence reducing the dangers related to DNS spoofing and illegal changes to DNS information.

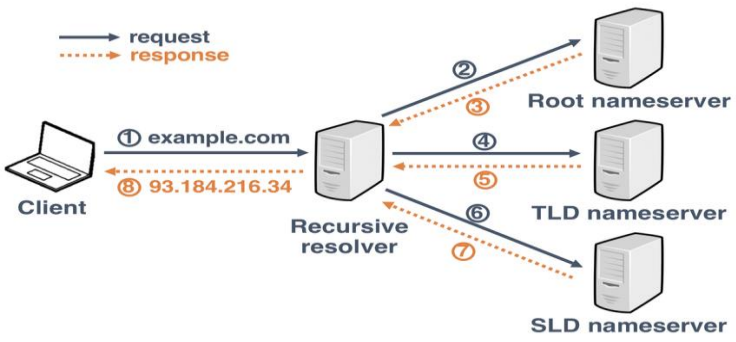


Figure 2 : Recursive Resolver

- **Root Nameservers**

As the first point of reference for DNS queries, root nameservers serve as the fundamental layer of the Domain Name System (DNS). Thirteen distinct companies oversee the strategically positioned root nameservers across the globe. These servers hold vital information about the DNS root zone but do not retain information about individual domain names. For every top-level domain (TLD), the DNS root zone keeps track of links to the authoritative nameservers. One of these root nameservers is consulted by a recursive resolver when it requires information about a TLD. Root nameservers send resolvers to the appropriate TLD nameservers in charge of the requested TLD, as opposed to giving out IP addresses for domain names directly. Because root nameservers point resolvers to the correct next tier of DNS servers in the hierarchy, this operation is essential to starting the DNS resolution chain and allowing it to progress methodically.

Because the root nameserver infrastructure is dispersed throughout the world, the DNS system is guaranteed to be resilient and redundant. Even though they only consist of 13 server clusters, they effectively manage a sizable amount of DNS traffic worldwide. These servers use anycast technology, which enables several physical servers to share an IP address and reply to requests according to which server is closest to the sender. By cutting down on latency and guaranteeing that queries are routed to the closest accessible root server, this configuration improves performance and reliability. The stability and dependability of these root servers, as well as the resilience of the DNS infrastructure that supports the operation of the entire internet, depend heavily on ongoing cooperation and monitoring between their administrators.

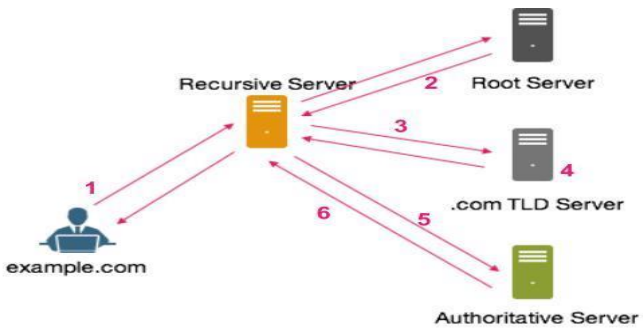


Figure 3 : Root Nameservers

- **TLD Nameservers**

Information relating to certain top-level domains, like.com,.org, and.net, as well as country-code TLDs like.uk,.de, and so on, is managed and kept up to date by TLD nameservers. Nameservers inside a TLD are responsible for storing important information about domains within that TLD. Recursive resolvers can obtain details about the authoritative nameservers in charge of certain TLD domains by contacting the nameserver for that TLD. For example, the appropriate TLD nameserver would point the resolver to the authoritative nameservers for a domain in the.com TLD when someone asks information about that domain. TLD nameservers are essential to DNS resolution because they direct resolvers to the authoritative nameservers, which are the next tier and contain exact information about specific domains.

TLD nameservers are spread among multiple sites and overseen by multiple entities, guaranteeing redundancy and dependability inside the DNS framework. They keep an accurate list of the authoritative nameservers for domains registered under each TLD in a directory that is updated on a regular basis. Because TLD nameservers act as a gateway to authoritative nameservers that store specialized domain-related data needed to appropriately resolve queries, their smooth performance is essential to the DNS's overall functionality.

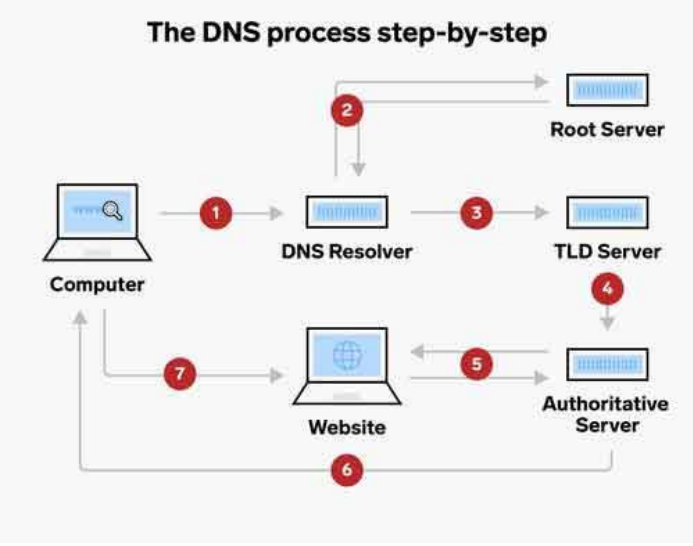


Figure 4 : TDL Nameservers

- **Authoritative Nameservers**

For a given domain, authoritative nameservers represent the last word on DNS information. They are in charge of maintaining and storing the actual DNS records for each domain, which also contain IP addresses, mail server details, and other domain-related data. The DNS resolution procedure is finished when a recursive resolver asks an authoritative nameserver for a specific domain and receives the needed data directly. The domain owners or administrators appoint these nameservers, and it is their responsibility to guarantee the availability and correctness of DNS data related to their domains. By precisely answering requests and routinely updating their records to reflect modifications made to domain setups, authoritative nameservers are essential to maintaining the integrity of DNS data.

The zones that authoritative nameservers oversee are divided into primary (master) and subordinate (slave) nameservers. The primary nameservers of a domain are the source of truth for its records and possess the original copies of the DNS zone data for that domain. Redundancy and fault tolerance are provided by secondary nameservers, which act as backups and replicate the data from primary nameservers. They are prepared to answer DNS queries in the event that the primary nameservers are down, and they synchronize with the primary nameservers on a regular basis to guarantee consistency. Maintaining the dependability of the DNS infrastructure and ensuring accurate resolution of domain names to their appropriate IP addresses depend heavily on the authoritative nameservers' smooth functioning.

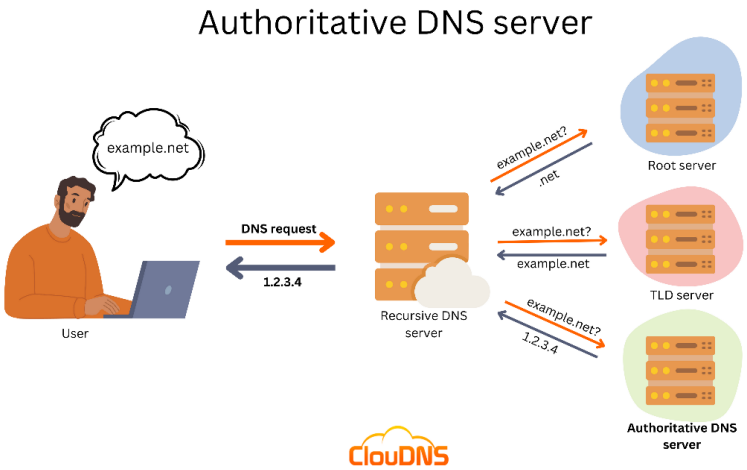


Figure 5 : Authoritative Nameservers

The way Domain name are organized

It is easy to find, categorize, and retrieve content from the internet because to the hierarchical and structured arrangement of domain names inside its framework. A variety of parts come together to form the Domain Name System (DNS), a crucial foundation that makes it possible to translate human-readable domain names into machine-readable IP addresses.

Top-Level Domains (TLDs)

Top-Level Domains (TLDs) are important parts of a domain name. Domains are grouped based on their function or geographic proximity.

- Generic Top Level Domains (gTLDs): Among them are popular extensions like.com,.org, and.net. At first, they served to identify the kind of website; for instance, a website ending in.com was meant to represent a business, a website ending in.org to represent an association, and a website ending in.net to represent a service linked to a network. With time, these distinctions have become less evident, and gTLDs are being used to achieve many goals.

- Nation-Code Top-Level Domains (ccTLDs): Associated with certain countries or areas, these comprise extensions such as.us,.uk,.de, and.fr. They provide as a visual indicator of the location of the website or organization.

- Sponsored Top-Level Domains (sTLDs) are domain extensions created with a particular industry or set of people in mind. Governmental organizations use.gov, academic institutions use.edu, the military uses.mil, and museums use.museum, to name a few. They guarantee a more focused classification of areas by offering services to certain sectors or groups.

Second-Level Domains (SLDs)

The unique symbols inside a specific TLD that follow top-level domains (TLDs) are known as Second-Level Domains, or SLDs. They typically act as a stand-in for the business, organization, or individual that owns the domain. For example, on google.com, "google" is the SLD. Using SLDs, businesses can develop a unique online identity that often corresponds with their brand names or specific products.

Sub-domains

Subdomains provide additional classification or differentiation inside an SLD. They appear before the primary domain name and allow the creation of distinct sections or services inside a domain. Subdomains are indicated as prefixes, separated by periods, from the primary domain name. The "mail" subdomain of "google.com," for instance, is how users are specifically directed to access the email service within the Google domain. Subdomains are essential for organizing websites and providing users with specific features or services within a broader domain hierarchy.

Protocols

Protocols, which provide the rules for data transmission back and forth between a web server and a web client, like a browser, are fundamental parts of URLs. For secure data transfer, they choose the data transmission protocols and set the encryption standards. "https://" indicates secure, encrypted communication, whereas "http://" indicates unsecured communication. The protocol employed in a URL ensures efficient and secure data transport over the internet, protecting sensitive information sent between users and servers.

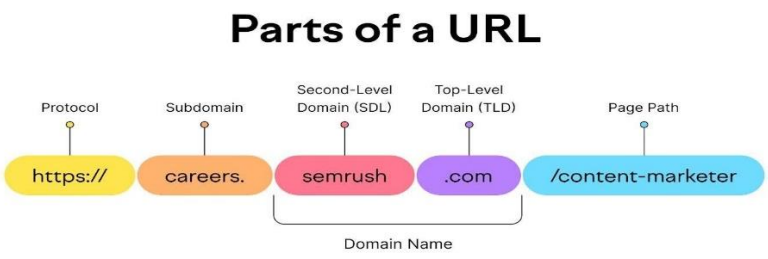


Figure 6 : The way domain name organize

Hierarchy of Domain Name Systems

The Domain Name System (DNS) hierarchy is an orderly framework that improves domain name resolution.

- The Root Domain

Denoted by a single dot (.), the Root Domain is the base level of the DNS hierarchy. For every domain name resolution, it acts as the foundation. It provides vital information that routes inquiries to the authoritative nameservers in charge of top-level domains (TLDs), even though it does not contain specific domain names. Consider it as the gateway that directs DNS servers, during domain name resolution, to the proper TLD nameservers.

- Top-Level Domains

Top-Level Domains are domains that come after the Root Domain. They can be of several types, including sponsored TLDs (sTLDs), country-code TLDs (ccTLDs), and generic TLDs (gTLDs). Every TLD category has a specific function. Originally intended for a limited range of entities, generic top-level domains (TLDs) such as.com and.org have expanded to allow for more applications. TLDs with country codes, like.us and.uk, designate certain regions or territories. Sponsored TLDs are specific to industries or communities; examples are.gov and.edu.

- Second-Level Domains (SLDs)

SLDs are the names that come directly after TLDs and act as special codes within a specific TLD. They are the primary component of a domain name and stand for companies, organizations, or other entities. For example, "google" is the Second-Level Domain in google.com. SLDs give organizations the ability to create a unique online identity within a category that the TLD has established.

- Subdomains

Within a Second-Level Domain, subdomains are divisions that come before the main domain name and provide additional organization or classification. They make it possible

to create unique services or parts inside a domain. In the case of blog.google.com, for instance, "blog" is a subdomain of "google.com," denoting a particular area or feature of the larger Google domain. Subdomains help to provide users with certain capabilities or information within a larger domain structure and help to organize websites in a systematic manner.

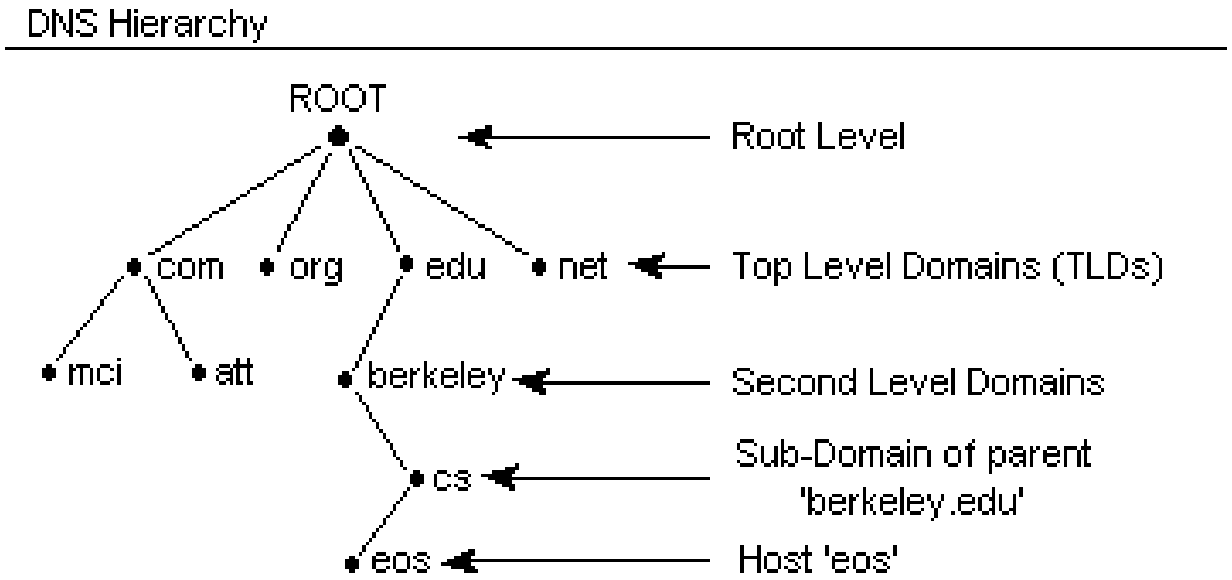


Figure 7 : DNS Hierachy

The purpose and relationships between communication protocols, server hardware, operating systems and web server software with regards to designing, publishing and accessing a website.

Communication Protocols

The fundamental collection of guidelines and practices that control data flow between systems or devices in a networked environment are known as communication protocols. They offer the structure for starting, stopping, and resuming communication sessions, guaranteeing dependable and smooth data transfer across a range of systems and technologies.These protocols, which lay the foundation for contemporary networking, specify the structure, synchronization, order, and error management of data transferred across devices. They enable devices to understand and interpret data transferred across networks by establishing the language spoken by various devices. Every protocol fulfills distinct purposes, meeting various demands related to networking and data transfer.

TCP/IP, UDP, SMTP, POP, FTP, HTTP, and HTTPS are among the protocols that are essential for enabling communication on local networks and the internet. They oversee several facets of the communication process, including address management, data integrity assurance, data packet splitting, and security measures.These protocols are important because they standardize communication, making it possible for systems and devices from many platforms and suppliers to communicate and share data without any problems. They serve as the foundation for contemporary digital communication, enabling the efficient operation of a vast worldwide network of networked devices for the purposes of email, file sharing, web browsing, and many other uses.

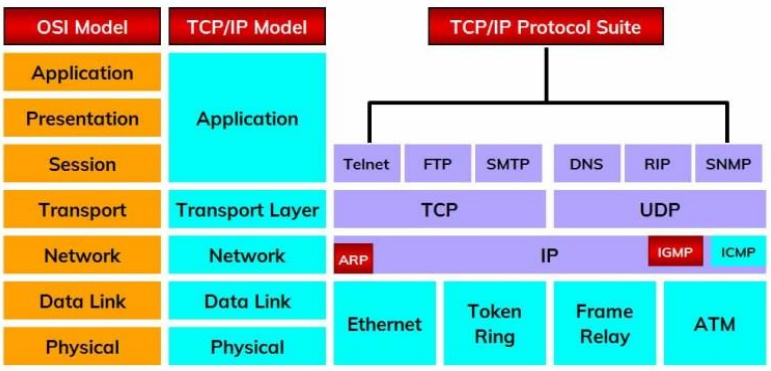


Figure 8 : communication protocols

Transmission Control Protocol (TCP)

Within the internet protocol suite, Transmission Control Protocol (TCP) is a fundamental communication protocol that guarantees error-checked, dependable, and well-organized data packet transmission between devices. It is connection-oriented, which means that it creates a connection before sending data and verifies that packets are delivered successfully by responding to the received ones. TCP uses techniques for error detection and retransmission in the event that packets are lost or corrupted. It also divides data into smaller components known as packets and provides sequence numbers for correct ordering. A three-way handshake is required to establish a connection, during which the sender and the recipient agree on the terms of data transmission. Data transfer happens in segments, each with a sequence number, after the link is made. In order to avoid congestion and packet loss, TCP additionally controls flow control, which makes sure that a sender doesn't send more data than a receiver can process.

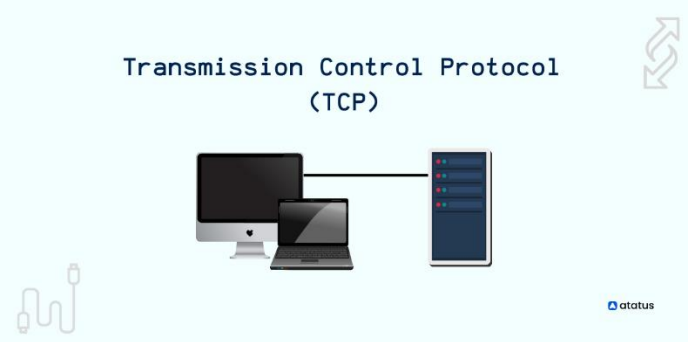


Figure 9:Transmission Protocol

Internet Protocol (IP) and IP Address

The foundation of internet communication is Internet Protocol (IP), which manages packet addressing and routing. It is in charge of logically sending data packets across networks from the originating device to the destination. Devices on a network are given numerical labels known as IP addresses, either IPv4 or IPv6. IPv6 employs a 128-bit address, which allows for unique device identification and communication, in contrast to IPv4's 32-bit address. To format, route, and transmit data packets to their proper locations, IP offers a set

of rules.The network and host components of IP addresses provide the necessary routing information. Devices can connect with one another and with other devices on the same network or the internet thanks to the allocations they make.

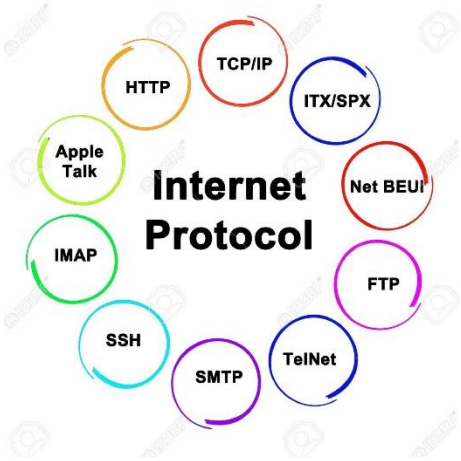


Figure 10 :Internet Protocols

Post Office Protocol (POP)

A common technique for getting emails from a mail server to a client device is called Post Office technique, or POP. POP enables email clients to download emails from the server using TCP/IP. POP comes in several forms, the most widely used of which being POP3. Emails are normally downloaded to the client device, and although settings can be changed to keep copies on the server, the default behavior is to erase the messages from the server after they have been downloaded.

‘

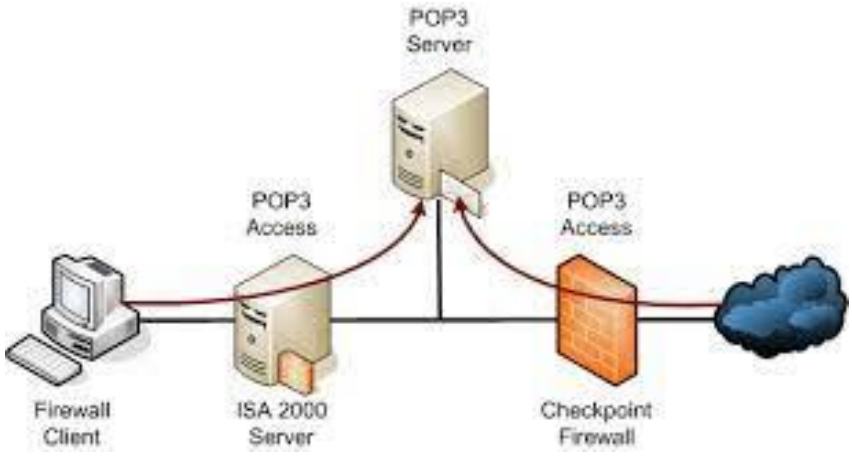


Figure 11 : Post office protocol

Simple Mail Transfer Protocol (SMTP)

Emails can be sent and received across networks using the Simple Mail Transfer Protocol (SMTP). It serves as a middleman for email transfers between mail servers. Emails are transmitted securely and reliably thanks to SMTP, which defines the guidelines for sending and receiving them, including message format.In order to distribute emails, SMTP uses a client-server architecture in which the sending client and receiving server interact. It uses a set of commands and replies to manage email sending, routing, and relaying. SMTP is in charge of ensuring that the message has been successfully sent to the recipient's mail server, transferring the message, and confirming the recipient's address.

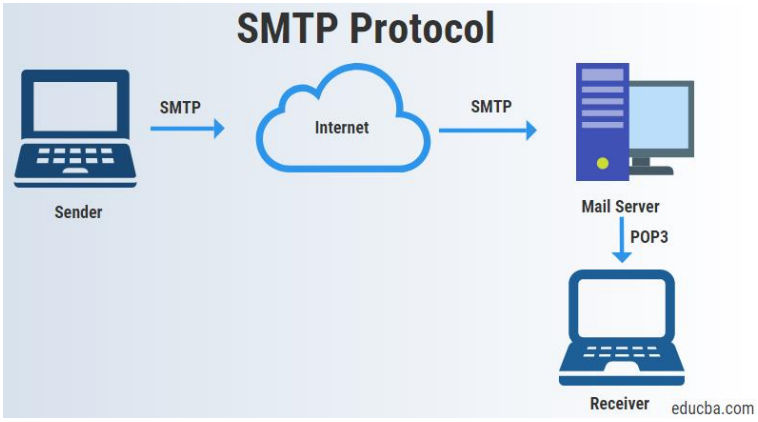


Figure 12 : Simple Mail Transfer Protocol

File Transfer Protocol (FTP)

On a computer network, files can be transferred between a client and a server using the File Transfer Protocol (FTP), a common network protocol. It enables users to handle files on a distant server and to upload and download them. FTP has distinct control and data connections and is based on a client-server architecture. The data connection is created when a file is actually transferred, whereas the control connection controls communication between the client and server.

FTP offers instructions for managing rights, renaming or deleting files, downloading files to the client, uploading files to the server, and navigating directories. Depending on how

the data connection is made between the client and server, it can function in either of two modes: active FTP or passive FTP.

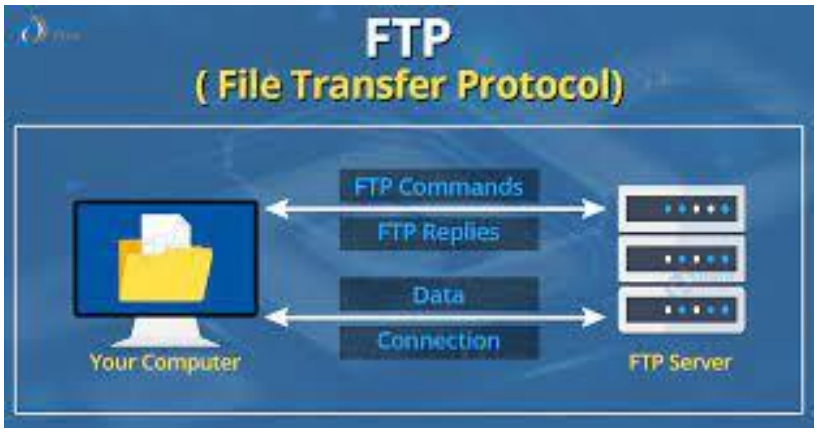


Figure 13 : File Transfer Protocol

Hypertext Trasnfer Protocol HTTP

The World Wide Web uses the Hypertext Transfer Protocol (HTTP) to transfer hypermedia documents like HTML files. It controls how a web client, or browser, and a web server communicate with each other. HTTP provides actions for web browsers and servers to take in conjunction with the formatting and transmission of messages.The way that HTTP works is that the client sends a request for resources (such web pages or photos) to the server, and the server responds by sending the requested data. To specify the activities to be taken, it makes use of methods like GET (retrieve), POST (submit), PUT (update), DELETE (remove), and others.

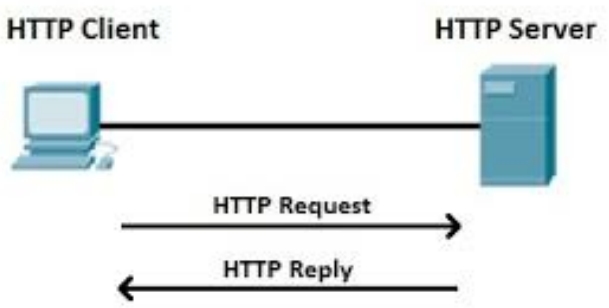


Figure 14 : Hypertext Transfer Protoco

User Datagram Protocol (UDP)

On the transport layer of the Internet Protocol suite, User Datagram Protocol (UDP) is a connectionless communication protocol. In contrast to TCP, UDP does not guarantee packet delivery or order and does not create a connection before transferring data. It is a lighter, quicker protocol that is perfect for uses where dependability is not as important as speed.When speedy transmission is desired above error checking and packet sequence maintenance, UDP is frequently employed for applications like online gaming, streaming media, video conferencing, and real-time communication where a tiny amount of packet loss can be accepted.

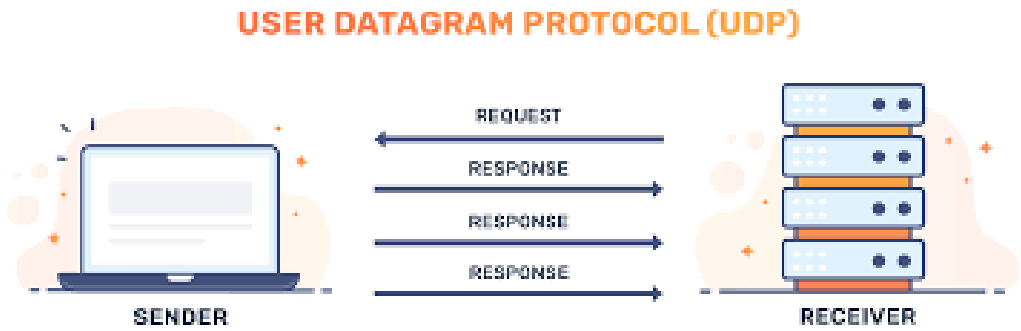


Figure 15 : User Datagram Protocol

Hypertext Transfer Protocol Secure (HTTPS)

A layer of encryption is added to HTTP in Hypertext Transfer Protocol Secure (HTTPS) to provide secure communication over a computer network, usually the internet. HTTPS encrypts data during transmission by using encryption protocols such as Secure Sockets Layer (SSL) or Transport Layer Security (TLS), which establish a secure channel between the client and serverSensitive data cannot be modified or intercepted while in transit when HTTPS is used. In order to establish safe connections and confirm the legitimacy of websites, it is dependent upon digital certificates that are issued by Certificate Authorities (CAs). A lot of people utilize HTTPS, especially when doing critical online banking, shopping, or visiting secure websites.

Server Hardware

Different configurations of server hardware are available, each tailored to fulfill particular requirements of performance, scalability, and space. Below is a summary of three typical server types:

Tower Servers

Tower servers are stand-alone devices with a design similar to desktop computer towers that are intended to be installed in server rooms or on top of desks. These servers' modest size and simplicity of configuration make them perfect for small to medium-sized enterprises or individual users. For individuals with minimal server infrastructure requirements, tower servers can be an affordable alternative because they are simple to operate and maintain.

Rack Servers

Rack servers are small, thin servers that are meant to be installed on racks in a server room or data center. Because of their stackability, many servers can be installed vertically in a rack enclosure to make optimum use of available space. Rack servers are a good option for companies that need a centralized and scalable computer infrastructure because of its easy-to-organize and scalable architecture. Hot-swappable parts are frequently included for simple maintenance.

Blade Servers

Compared to rack servers, blade servers are smaller and more highly modular servers. They are made up of separate server modules called "blades" that fit into a common chassis or enclosure. Blade servers provide high-density computing by enabling the sharing of resources such as power supplies, cooling, and networking across several blades housed in a single enclosure. They are perfect for businesses where power and space efficiency are critical and high-density computing is required.

Types of servers

File servers

File servers are specialized devices made to handle and store data centrally over a network. They offer a centralized site where network users can store data, media files, and documents. These servers guarantee data consistency and offer a single platform for data administration, making file sharing, access control, and collaborative work settings easier.

print servers

Print servers oversee and handle network-wide printing jobs. They control printer access, monitor print queues, and maximize printer utilization. By effectively handling print jobs, controlling several printers, and providing centralized print management features, these servers improve print resource management.

Application servers

Application servers offer a platform for the deployment, operation, and upkeep of software applications by hosting and managing those applications. Scalability, performance, and efficient application deployment inside an organization's infrastructure are guaranteed by them as they manage application logic, data processing, and user access to apps.

DNS servers

Human-readable domain names are converted into IP addresses by DNS servers, also known as Domain Name System servers. In order to allow users to access websites and other internet resources using domain names, they maintain databases that translate domain names to matching IP addresses. DNS servers are essential for redundancy and effective internet traffic routing, which ensures dependable access to web services.

Mail servers

Emails are sent, received, and stored by mail servers, which also handle email communications. They manage email transfer and storage technologies, such as SMTP and POP/IMAP. To provide safe and effective email communication, these servers include security features including spam filtering, authentication, and encryption.

Web servers

Web servers are specialized hardware or software that respond to requests from clients via the internet by serving web content. They handle internet traffic, process HTTP requests, and deliver online pages. These servers use protocols like HTTP and HTTPS to transfer data between servers and clients in order to host websites, apps, and other web-based services. They process, store, and make content available to users globally, including photographs, videos, and web pages

Database servers

Database servers handle requests to access, add, edit, and remove data from databases as well as maintain and store structured data. They offer a centralized location for effective data management and storage. These servers organize, secure, and retrieve data in response to queries using database management systems (DBMS) such as MySQL, Oracle, or SQL Server.

Virtual servers

While operating inside a physical server, virtual servers are separate virtual machines (VMs) with their own operating system and software. Multiple virtual servers can operate on a single physical server thanks to virtualization technology, which maximizes resource usage and boosts flexibility. These servers provide efficient resource allocation depending on different workloads, improve scalability, and enable server consolidation.

Proxy servers

By serving as a middleman between clients and other servers, proxy servers enable oblique connections over a network. They take requests from clients, route them to the intended servers, and then provide the results to the clients. By caching data, screening queries, and acting as firewalls to ensure restricted access to resources and defend against outside threats, proxy servers improve security, privacy, and performance.

Monitoring and management servers

In order to guarantee the dependability, security, and efficiency of network operations, monitoring and management servers supervise and regulate them. They gather and examine information from different servers, network devices, and apps in order to monitor setups,

find problems, and maximize resource use. These servers frequently run specialist software that gives administrators access to tools and insights for efficient system management and troubleshooting. This software is used for monitoring, logging, and controlling network infrastructure.

Server operating systems

An operating system (OS) is an essential software layer that controls hardware resources and offers a computing device's application platform. Through the effective allocation of resources like CPU, memory, and storage, it makes communication between users and the computer's hardware possible. Operating systems provide a graphical, command-line, or touch-based user interface that lets people communicate with computers. They oversee memory, file systems, and processes to guarantee that several programs run smoothly at once. To protect data and allow devices to connect over networks, networking and security characteristics are combined. Popular OS choices include mobile operating systems like Android and iOS, Microsoft Windows, macOS, Linux, and Unix, each designed for certain devices and user requirements.

Figure 16 : Server operating systems

Microsoft Windows servers

Microsoft created Windows Server, an operating system intended for use on servers. It is well-liked in a variety of business contexts due to its user-friendly interface, robust support, and numerous applications. Strong security features, network resource management through active directory services, and interoperability with Microsoft apps like SharePoint

and Exchange Server are all provided by Windows Server. In order to meet a wide range of corporate demands, it offers multiple server roles, including file and print services, web hosting, and database management.

Figure 17 : Microsoft Windows server

Linux / Unix servers

Operating systems based on Linux and Unix are renowned for their dependability, security, and adaptability. They provide a wide range of distributions (including Ubuntu, Red Hat, and CentOS) that are ideal for server installations. Because of their strong security features and potent command-line interfaces, they are well-liked by system administrators, developers, and businesses. Because of these systems' great degree of customization, users can create setups specifically for web hosting, database management, or file sharing on servers. Servers running Linux and Unix provide superior performance, dependability, and scalability.

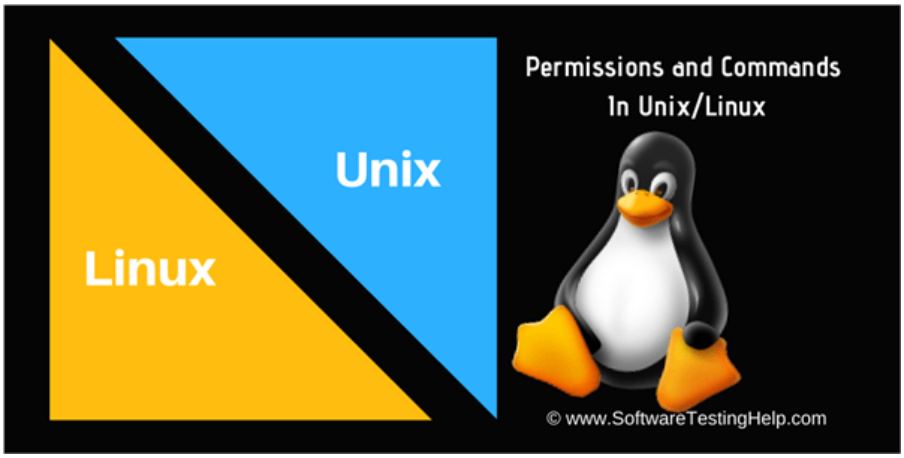


Figure 18 : Linux / Unix Servers

Oracle Solaris

Oracle Corporation created Oracle Solaris, an operating system based on Unix. Mission-critical enterprise applications can benefit from its sophisticated features, scalability, and reliability. High security, resource management, and broad support for virtualization technologies are all offered by Solaris. It is preferred in sectors like finance, telecommunications, and large-scale data centers where uptime, performance, and security are critical.

Figure 19 : Oracle Sloaris

Mac OS X Server

Developed by Apple Inc., Mac OS X Server is an operating system based on UNIX that is intended for Macintosh computers and servers. Along with tight integration with Apple devices, an easy-to-use interface, and a range of server features including file sharing, email services, and network management are all included. Mac OS X Server is popular in the creative industries, academia, and enterprises with Mac-centric settings, even though it may not be as common in enterprise situations as other server OS solutions. It offers seamless connectivity with other Apple devices and applications.

Figure 20 : Mac Os X server

Web Server Software

Web server software is the foundation for supplying users' browsers with web content. Using a client-server architecture, it accepts and responds to HTTP requests from web browsers. These servers can manage dynamic content generation using programming languages like PHP, Python, or Ruby, or they can handle static material like HTML, pictures, and multimedia files.

Among the most popular and established HTTP servers, Apache provides cross-platform interoperability, a large module support base, and strong functionality. High performance, effectively providing static content, and managing concurrent connections are the main goals of Nginx. With its smooth integration with Windows environments, Microsoft's Internet Information Services (IIS) offer web applications developed using.NET and ASP.NET technologies a strong platform.
These servers guarantee effective request processing, security via SSL/TLS encryption, load balancing for heavy traffic, and frequently incorporate caching algorithms to maximize the speed at which material is delivered, so enhancing the user experience.

Application Server Software

Serving as a middleware platform, application server software makes it easier for apps to run. It offers a setting in which application components can operate, handle security, handle transactions, and integrate with databases. These servers include with capabilities like connection pooling, clustering, and failover support, and they support a wide range of programming languages and frameworks.

The Apache Software Foundation created Tomcat, which is optimized for Java-based servlets and web applications. A stable Java EE environment is provided by Red Hat's open-source JBoss application server. Comprehensive enterprise-level application server solutions that support numerous programming languages and offer scalability and dependability are offered by IBM's WebSphere and Oracle's WebLogic.
These servers are essential for handling business logic, deploying and maintaining web applications, and facilitating communication between different parts of a software stack.

Database Server Software

This software controls how data is stored, retrieved, and arranged in databases. They provide effective data management by offering features like data integrity, security, and query processing. A feature-rich relational database management system (RDBMS) with scalability, security, and support for intricate transactions is Oracle Database. MySQL is a well-known open-source database that is perfect for online applications due to its speed, dependability, and ease of use. Microsoft SQL Server offers powerful capabilities and business intelligence tools, and it interfaces with Windows systems with ease.

Cloud Computing Server Software

Cloud computing server software is the primary component of cloud infrastructure, providing the virtualized resources and online services that are supplied. In a cloud setting, this software makes it possible to provision, manage, and orchestrate virtual machines, storage, networking, and other resources.Users of the open-source cloud computing platform OpenStack can oversee resource pools using a dashboard interface. It offers flexibility and scalability and supports both private and public clouds. One of the best proprietary solutions available is VMware vSphere, which provides powerful cloud computing and virtualization features for effective resource management and use.In order to meet the varied needs of users, these solutions include scalability, on-demand resource allocation, and many service models such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).

File server software

Within a network, file server software allows file sharing and data access while also managing centralized storage. It offers a safe and well-organized method of data storage by enabling users to save, retrieve, and share files amongst devices and users on the same network.Microsoft's popular file server solution, Windows Server, comes with features including user authentication, access control, and easy integration with Active Directory. Linux-based solutions, such as Samba, allow efficient and affordable file sharing across

heterogeneous systems by providing file and print services that are compatible with Windows environments.These servers improve cooperation, data accessibility, and security within enterprises by guaranteeing data integrity, access control, and data backup. In order to simplify file administration and access, they also provide capabilities like versioning, rights management, and quota control.

The impact of common web development technologies and frameworks with regards to website design, functionality and management.

Web development technologies and frameworks

- HTML(Hypertext Markup Language)

The basis of the World Wide Web is HTML (HyperText Markup Language), which is the language used to create web pages. It offers the fundamental organization and design for online information. HTML annotates elements by describing their structure and how they show up in web browsers through a system of tags. These components consist of lists, headings, paragraphs, pictures, and links, among others.
One of HTML's primary characteristics is its accessibility and simplicity, which make it simple for newcomers to learn and utilize. It works on a simple idea of using a set of predefined tags to structure content. These tags outline the relationships between various elements of a webpage and are arranged hierarchically.

HTML defines the meaning and accessibility of web content in addition to its visual aspects. Developers can use HTML to provide meaningful content that is accessible to a wide range of users, including those who use assistive technologies. It guarantees that information will show consistently across various browsers and devices and that search engines can understand it.HTML is always evolving along with the web. New features and capabilities were added with HTML5, its most recent version, including native support for multimedia elements, improved form functionality, and improved support for mobile devices. Richer user experiences are made possible by the web being far more interactive and engaging thanks to HTML5.

HTML is essentially the building block that supports the creation of cohesive, orderly, and user-friendly web pages by offering the structure, semantics, and accessibility that are required.

Advantages and Disadvantages of HTML

Advantages	Disadvantages
• **Simple Syntax**	• **Limited style**
Beginners will find it easy to understand and utilize due to the syntax's simplicity.	Requires CSS for advanced design; lacks complex style features.
• **Cross-Platform Compatibility**	• **Static Nature**
HTML is compatible with a wide range of browsers and platforms, making it universally accessible.	Produces static webpages devoid of dynamic content or interaction.
• **SEO friendliness**	• **Browser Compatibility**
HTML with proper structure improves search engine ranks.	If not standardized, may display differently in different browsers.
• **Versatility**	• **Security flaws**
It can include a variety of media formats, including audio, video, and image embeds.	If sufficient validation isn't done, it can be vulnerable to attacks like XSS.
• **Widely Acknowledged Standard**	• **Lack of Control**
This well-established standard guarantees interoperability among various web technologies.	Compared to design-focused tools, there is less control over the accuracy of the layout.

• **Semantic components**	• **Overuse of Tags**
provided by accessibility features to support screen readers and other assistive technology.	Using too many needless tags can clog code and make it harder to read.
• **Offline Access**	• **Semantic Misuse**
Web apps can function without an internet connection thanks to HTML5's support for offline data storage.	When tags are misused, erroneous or deceptive document structures may result.
• **Flexibility**	• **Limitations of SEO**
HTML is used by responsive design frameworks to generate mobile-friendly web pages.	Without additional tactics, optimizing rankings just using HTML may not be possible.

Table 1 :Html advantages and Disadvantages

Useres of Html

The primary language used to create web information is HTML (HyperText Markup Language). Its main function is to give web pages their structural framework. HTML establishes the foundation on which websites and applications are constructed by specifying the layout, content hierarchy, and fundamental elements.HTML is used extensively on websites to describe elements like lists, headings, paragraphs, and images. It plays a crucial role in providing content with semantic meaning and organization. For example, using certain tags to distinguish between the primary text, headers, footers, and navigational components, developers can create the structure of a webpage using HTML.

HTML serves as the foundation for the addition of CSS (Cascading Style Sheets) and JavaScript, which improve the look and feel of online pages. Its application goes beyond

adding multimedia components, such as pictures, videos, and audio, to offer a comprehensive browsing experience.HTML is also essential for building web forms, which allow you to capture user data via input fields, checkboxes, radio buttons, and submit buttons. The versatility of HTML5 is demonstrated via responsive web design, wherein the properties of the HTML5 language enable the development of mobile-friendly websites that conform to varying screen dimensions.All things considered, HTML serves as the foundation of the internet, enabling accessibility, content management, and the fundamental structure of web pages on a wide range of hardware and browsers. Because of its semantic nature and ease of use, it is essential for creating a unified and approachable online presence.

Figure 21 : hypertext Markup Language(HTML)

CSS (Cascading Style Sheets)

A styling language called CSS (Cascading Style Sheets) is used alongside HTML to specify how web pages are presented and laid out. It gives developers the ability to specify colors, fonts, spacing, placement, and other visual features of a website, giving them control over how HTML elements look.

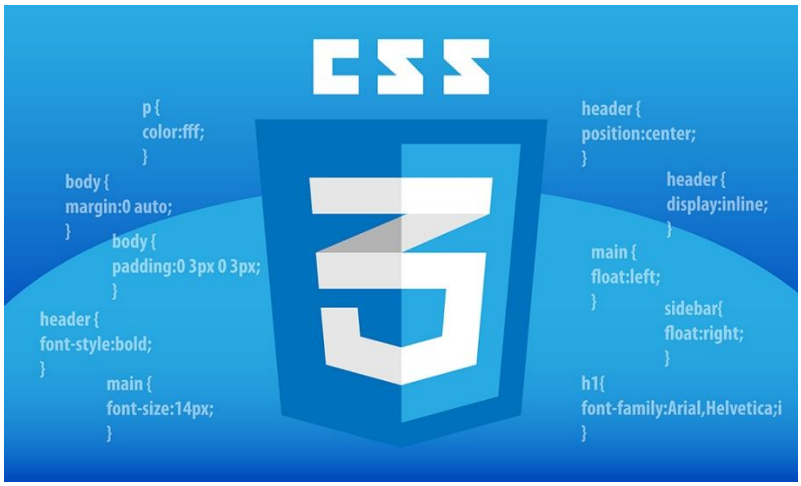


Figure 22 : CSS (Cascading Style Sheets)

Advantages and Disadvantages of CSS

Advantages	Disadvanatges
Content and Presentation Separation	**Browser Compatibility**
By separating a webpage's content from its visual presentation, CSS improves maintainability and makes upgrades simpler.	Rendering on different platforms may be inconsistent due to disparities in how different browsers interpret CSS rules.
Consistency and Reusability	Complexity in Big Projects: Keeping up with several style rules and managing CSS in larger projects can get difficult.
CSS makes it possible to create style guidelines that may be used on different	

pages, guaranteeing that a website's design elements are consistent from page to page.	**Learning Path**
Faster Loading	For newcomers, CSS can have a high learning curve, particularly when combined with sophisticated layout techniques.
Browsers have the ability to cache external CSS files, which speeds up the loading of websites on recurrent visits.	**Limitations**
CSS provides methods such as media queries that make responsive design possible, enabling websites to adjust to various screen sizes and devices.	It may be difficult to accomplish some design features with CSS alone, calling for other technologies or workarounds.
Accessibility	**Specificity and Inheritance**
Developers can improve usability by using CSS to build designs that are suitable for users with disabilities.	If not correctly maintained, CSS's specificity rules and inheritance hierarchy may cause unforeseen styling problems.
SEO-Friendly	**Performance Impact**
By separating presentation from information, web pages are easier for search engines to index and rank higher.	The speed at which a page renders might be affected by using complicated selectors or excessive CSS usage.

Table 2 : Advantages and Disadvantages of CSS

Uses of CSS

- Styling Web Pages: CSS is mostly used to specify how elements should be displayed in HTML texts.
- The creation of responsive layouts that adjust to different screen sizes and devices requires responsive design.
- Animations and Effects: JavaScript is not required to create animations, transitions, or effects with CSS.
- Consistent Branding: This refers to keeping a website's branding and design aspects constant throughout.

With its ability to design webpages that are aesthetically pleasing, responsive, and accessible, CSS is essential to modern web development. Its strength lies in its ability to keep display and content separate, which promotes efficiency in design, consistency, and reuse. However, there are issues with complexity, compatibility with different browsers, and performance concerns.

Programming Languages

PHP

PHP is a popular server-side scripting language that was first used for creating websites. PHP was originally designed to manage personal homepages, which is why the abbreviation came about. Today, it is a flexible language that can be used to develop dynamic websites and web apps. Because of its smooth HTML integration, developers can insert PHP code straight into HTML pages. PHP powers a large chunk of the internet with its many frameworks and content management systems, like as WordPress, making it easier to create dynamic and interactive web experiences.

Figure 23 :PHP

Advantages and Disadvantages of PHP

Advantages	Disadvantages
• Simple to Learn: PHP is easily mastered by newcomers due to its mild learning curve.	• Security: PHP had security flaws in the past, but security has been strengthened in more recent versions.
• Wide Adoption: It is the engine behind a large chunk of the web and offers a wealth of information and community assistance.	• Performance: PHP may not be as performant as some other languages.
• Integration: PHP is appropriate for web development and the creation of dynamic content since it integrates with HTML with ease.	• Inconsistencies: Because of its flexible and loosely typed syntax, PHP may have inconsistent behavior in huge codebases.
• Open Source: PHP's open-source nature makes it free to use, which lowers the cost of development.	
• CMS and Frameworks: It features well-known content management systems like WordPress and Drupal together with strong frameworks like Symfony and Laravel.	
• Simple to Learn: PHP is easily mastered by newcomers due to its mild learning curve.	

Table 3 : Adavantages and Disadvantages of PHP

Uses of PHP

- Server-side scripting
- web development
- creating dynamic webpages
- e-commerce platforms

- content management systems
- web applications are the main applications for PHP.

JavaScript

JavaScript is a high-level interpreted scripting language that is mostly used for client-side web development. It is sometimes referred to as the "language of the web." It was once created to provide dynamic behavior to web pages, but with the help of Node.js, it has developed into a flexible language that can operate on the client and server sides. JavaScript manages user events, modifies the DOM, and permits asynchronous communication to make websites interactive. Single-page applications (SPAs), front-end frameworks, and server-side apps may all be developed more easily because to its vast ecosystem of frameworks and libraries.

Figure 24 :Javascript

Advantages and Disadvantages of Javascript

Advantages	Disadvantages
• JavaScript's versatility stems from its ability to be used not only for front-end development but also for server-side programming via Node.js.	• Browser Compatibility: It can be difficult to guarantee consistent functionality across many browsers.
• Interactive and Dynamic: It allows web pages to have animations, dynamic content, and interactive elements.	• Security Risks: Because it is a client-side language, improper handling might lead to security risks.

• A vast array of frameworks and modules, including as React, Vue.js, and Angular, augment its capability inside this vast ecosystem.	• Performance: Because JavaScript is single-threaded, it may be affected by large computations.
• Asynchronous programming is made possible by JavaScript's event-driven structure, which enhances the responsiveness of online applications.	• Single Threaded: JavaScript is single-threaded, meaning it can execute only one operation at a time within a process. This might cause performance bottlenecks when handling multiple operations simultaneously.
• Form validation and user experience enhancement are two common uses for client-side validation.	• Debugging Challenges: Debugging JavaScript can be challenging, especially in complex applications. Dynamic typing and loose syntax can sometimes lead to harder-to-find bugs.
• JavaScript's versatility stems from its ability to be used not only for front-end development but also for server-side programming via Node.js.	

Table 4 : Adavantages and Disadvantages of javascript

Uses of javascript

- It is an essential component of web development that makes websites' dynamic and interactive features possible. It is employed in the creation of responsive user interfaces, animations, dynamic content updates, and form validation.

- Front-End Development: JavaScript powers the front end of online applications, working in tandem with HTML and CSS. JavaScript is used by frameworks like React, Angular, and Vue.js to create intricate and dynamic user interfaces.

- Back-end development: Using the JavaScript runtime Node.js, programmers may create server-side JavaScript apps. It makes full-stack development possible by allowing programmers to create front-end and back-end applications using JavaScript.

- Mobile App Development: JavaScript is used to create cross-platform mobile applications that function flawlessly on a variety of devices. Frameworks such as React Native and those that enable Progressive Web Apps (PWAs) are examples of this.

- JavaScript is utilized in game development, particularly for mobile and browser-based games. Web browser-based 2D and 3D game development is facilitated by libraries such as Phaser and Three.js.

- Web servers and APIs: Web servers and APIs are built with JavaScript. A well-known Node.js framework called Express.js makes server-side scripting easier and aids in the creation of reliable APIs.

Python

Python is a popular high-level interpreted programming language that is easy to learn and has a lot of adaptability. Both novice and experienced developers will find it to be a great option due to its readable style and simple syntax. Web development, data analysis, artificial intelligence, scientific computing, automation, and other diverse applications are supported by Python's vast standard libraries as well as third-party frameworks like Django and Flask. Its widespread success across several fields has been greatly influenced by its simplicity of use as well as the presence of a robust and encouraging community.

Figure 25:Python

Advantages and Disadvantages of Python

Advantages	Disadvantages
• Simplicity and Readability: Python is simple to read and understand because of its brief, unambiguous syntax, which closely mimics that of the English language. This readability increases productivity and lowers program maintenance costs.	• Speed Restrictions: Python executes more slowly than languages of a lower level, such as C or C++. This is because dynamic type and its interpreted nature can lead to decreased performance on some high-computational applications.
• Versatility: It is compatible with procedural, object-oriented, and functional programming paradigms, among others. Python is appropriate for a wide range of applications, including artificial intelligence, scientific computing, and web development.	• Global Interpreter Lock (GIL): The Global Interpreter Lock, which only permits one thread to execute Python bytecode at a time, can limit the execution of multiple threads concurrently in multi-threaded Python programs, affecting performance in CPU-bound jobs.
• Huge Libraries and Frameworks: NumPy, Pandas, TensorFlow, Django, Flask, and many more are just a few of the many libraries and frameworks available for Python. By making complicated operations simpler, these tools speed up development and lessen the need to write code from scratch.	• Limitations of Mobile Development: Compared to languages like Swift for iOS or Kotlin for Android, Python is less common in the development of mobile apps. While frameworks like as Kivy and BeeWare do exist, they might not provide every feature found in native development languages.
• Community and Support: The Python community is quite huge, vibrant, and helpful. It provides a wealth of tutorials, user forums, and documentation to support learning, problem-solving, and teamwork.	• Design Restrictions: Because Python is dynamic, there may be fewer compile-time errors, which could mean that possible mistakes are only found during runtime. Large-scale projects

	may find this challenging and require more thorough testing.
• Scalability: By utilizing its helpful frameworks and libraries, Python can handle growing workloads and remains scalable. It effectively powers large-scale applications as well as small-scale scripts.	• Memory Consumption: Because of its high-level structures and dynamic typing, Python uses more memory than certain other languages. When working with really large-scale systems or in situations with memory constraints, this could be a disadvantage.
• Integration Skills: It easily integrates with various systems and languages. Because of its flexibility, Python can be easily integrated with C, C++, Java, and other languages, making it more platform-neutral.	• Packaging Issues: Although Python's package management system, which makes use of tools like pip, is reliable, maintaining a stable environment can occasionally be a little difficult due to conflicts arising from handling dependencies and versions across programs.

Table 5 : Adavantages and Disadvantages of python

Uses of Python

- Web development: Scalability, security, and resilience are provided by Python frameworks such as Django and Flask, which facilitate the quick creation of web applications. Businesses like Pinterest (Flask) and Instagram (Django) use it for their backend services.

- Data science and machine learning: Because of its ease of use and extensive ecosystem of specialized libraries, Python—along with libraries like NumPy, Pandas, and Scikit-learn—is the preferred language for data analysis, visualization, and machine learning. Popular Python-based deep learning libraries include PyTorch and TensorFlow.

- Scientific Computing: Python is useful for scientific computations, simulations, and research because of its scientific libraries. Its capabilities are extended by scientific tools such as SciPy and Matplotlib, and Python is used in astronomy, biology, physics, and other fields.

- Automated Testing and Scripting: Python is a great language for scripting and automation jobs because of its easy-to-read syntax. It is extensively utilized in network automation, system administration, and test script development.

- Backend Development: A lot of server-side scripting, API creation, and server-side logic is done with Python. It supports millions of users' daily experience on platforms like Spotify, Dropbox, and YouTube.

- Desktop GUI apps: Tkinter, PyQt, and Kivy are just a few of the frameworks that Python supports GUI development with, making it simple for developers to construct cross-platform desktop apps.

Ruby

Ruby is a well-known, high-level interpreted programming language that is easy to learn and understand. Ruby is a productivity and developer-focused language that prioritizes convention over configuration and has a clean, simple syntax. With the release of Ruby on Rails (RoR), a potent web application framework that simplifies and accelerates web development, it became incredibly popular. Because of Ruby's simple syntax and the opinionated structure of the Rails framework, developers can create reliable and manageable web apps quickly.

Figure 26 : Ruby

Advantages and Disadvantages of Ruby

Advantages	Disadvantages
• its simple and readable syntax, which emphasizes productivity and simplicity while cutting down on development time.	• Performance: Because Ruby is an interpreted language and has dynamic feature overhead, it may be slower than some other languages. This can have an effect on performance in high-load applications.
• Strong Community Support: It has a large and active community that adds to a multitude of frameworks, libraries, and tools, accelerating and streamlining development.	• Challenges with Scalability: Although RoR works great for developing prototypes and small- to medium-sized applications, scaling it up to enterprise-level systems may be difficult if proper optimization isn't done.
• Ruby on Rails (RoR): Rails is a robust framework for web applications for Ruby that simplifies web development duties with built-in tools, conventions, and a large ecosystem.	• Learning Curve: The versatility of the language can result in intricate and unusual coding techniques, which may be difficult for beginners to understand at first.
• Flexibility: Ruby is an extremely versatile language that lets programmers write code in procedural, functional, or object-oriented paradigms, among other approaches.	• Concurrency: Ruby's built-in support for concurrency is not as strong as that of certain other languages, which makes it less appropriate for intensively concurrent, CPU-bound work.
• Dynamic Typing and Reflection: These features provide greater coding flexibility by simplifying the creation of abstract and reusable components.	• Interoperability: Ruby is adaptable, but it may need extra work or workarounds to integrate with certain legacy systems or third-party tools.

Table 6 : Adavantages and Disadvantages of Ruby

Uses of Ruby

- online development: Ruby on Rails is a popular choice for startups and businesses looking for quick development and prototyping because it powers a large number of online apps.

- API Development: Because Ruby frameworks like Grape are so user-friendly and straightforward, they are frequently utilized to create RESTful APIs.

- Automation and Scripting: Ruby is a good choice for building scripts for a variety of uses due to its readable and simple syntax.

- Education & Training: Ruby is a popular language for novices and instructors imparting programming topics because of its simplicity and emphasis on readability.

- Backend Services and Systems: Because of Ruby's strong frameworks and ease of development for certain use cases, many businesses use it for backend services and systems.

Frameworks

Node.js

Based on the V8 JavaScript engine seen in Chrome, Node.js is a JavaScript runtime. It enables server-side scripting and the creation of scalable network applications by enabling developers to run JavaScript code outside of a browser.

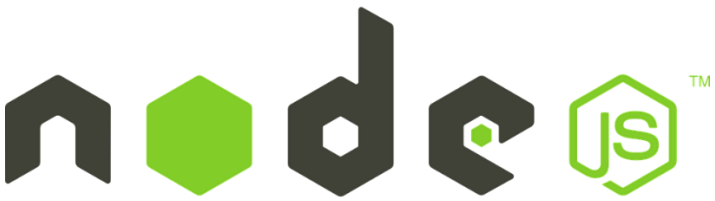


Figure 27 : Node.js

Advantages and Disadvantages of Node.js

Advantages	Disadvantages
• High Performance: Node.js is a fast and appropriate framework for real-time applications since it operates on an event-driven, non-blocking I/O model.	• Nestled callbacks can result in convoluted code architectures and "callback hell."
• Scalability: Because it doesn't stall, it can easily scale and is appropriate for processing multiple requests at once.	• APIs that are unstable: Node.js is evolving so quickly that it can occasionally lead to unstable APIs that need to be updated and adjusted often.
• Huge Ecosystem: It has an extensive library of modules for a variety of functions that are accessible using npm (Node Package Manager).	• Limited CPU-Intensive jobs: Because Node.js is single-threaded, it might not be the ideal option for CPU-bound jobs.
• One Language: The development process is streamlined when JavaScript is used for both client-side and server-side programming.	• Absence of Built-in Support: In certain situations, the need for third-party modules to perform specific functions increases dependence.
• Community Support: It has a sizable and vibrant community that offers a wealth of tools and resources to support its expansion.	• Learning Curve: Node.js might have a high learning curve for those who are not familiar with the asynchronous programming paradigm of JavaScript.

Table 7 : Adavantages and Disadvantages of Node.js

Uses of Node js

- Instantaneous Applications: Node.js is a great tool for creating live-tracking applications, gaming servers, and chat applications.

- API Development: Because of its scalability and non-blocking I/O, it is frequently used to create RESTful APIs.
- Microservices: Small, independently deployable services can be created with Node.js.

- It is used in the development of Single Page Applications (SPAs) to facilitate seamless client-server interactions.

- IoT Applications: Because of its lightweight and event-driven architecture, Node.js is becoming more and more popular in IoT projects.

Rails on Ruby (RoR)

Written in Ruby, Ruby on Rails is a powerful, open-source web application framework. Operating on the tenet of "convention over configuration," it provides default web page, web service, and database architectures.

Figure 28 : Rails on Ruby

Advantages and Disadvantages of Ruby on Rails

Advantages	Disadvantages
• Prototyping and development may happen quickly because to RoR's convention-based structure.	• Performance: Because it is interpretable, performance may be impacted by intricate calculations and demanding jobs.
• Libraries and Community: For more functionality, a vibrant community adds gems, or plugins.	• Learning Curve: It may take some time for novices to comprehend and become proficient with the norms and principles.
• Active Record: Object-Relational Mapping (ORM) in RoR streamlines	• Updates and Dependencies: Version conflicts or deprecated code may arise

data handling by simplifying database interfaces.	from frequent updates and gem dependencies.
• Features of Security: Application security is improved by built-in security features like SQL injection prevention and CSRF protection.	• Challenges with Scaling: Although RoR is scalable, managing large numbers of concurrent connections could be difficult.
• Scalability: RoR is suited for expanding organizations since it makes it simple to scale and modify applications.	• Monolithic Nature: It might be difficult to expand and manage large, monolithic applications over time.

Table 8 : Adavantages and Disadvantages of Ruby on Rails

Uses of Ruby on Rails

- Web apps: From small businesses to major corporations, RoR is widely utilized for developing web apps.

- Content Management Systems (CMS): RoR's adaptability and user-friendliness make it a valuable component of many CMS platforms.

- E-commerce: RoR's quick development features enable a number of e-commerce platforms.

- API development is used to create reliable backend systems and APIs.

- Prototyping: RoR is well-known for producing MVPs (Minimum Viable Products) and prototypes fast and effectively.

Bootstrap

Developed by Twitter, Bootstrap is a front-end framework that provides a set of HTML, CSS, and JavaScript elements. With cross-browser interoperability and responsive design, it's made to simplify web building.

Figure 29 : Bootstrap

Advantages and Disadvantages of Boostrap

Advantages	Disadvantages
• responsive design: Bootstrap makes sure websites adjust fluidly to different screen sizes and devices.	• Similar Look: Due to the widespread usage of Bootstrap, websites created using it may appear similar in terms of design out of the box.
• Pre-designed Components: It cuts down on development time by offering a library of pre-designed components.	• download Size: Loading times for pages may be impacted if the full framework is included in the download.
• Cross-Browser Compatibility: Bootstrap's elements work with most major browsers, guaranteeing layout consistency.	• Learning Curve: For novices, getting the hang of Bootstrap's many features and classes may take some time.
• Customizable: It may be made more unique by adding extensions, extra JavaScript plugins, and themes.	• Overrides and Specificity: Because of the specificity of CSS, customizing default styles can be difficult.

• Community and Documentation: To facilitate adoption, Bootstrap boasts a large community and thorough documentation.	• Dependency on JavaScript: Without JavaScript, some components' functionality may be reduced, which could affect speed.

Table 9 : Adavantages and Disadvantages of Boostrap

Uses of Boostrap

- Bootstrap is great for building responsive websites, which change fluidly to fit different screen sizes.
- Web application prototype can be done fast using it because to its pre-designed components.
- CMS Themes: A variety of Content Management Systems can have their themes created using Bootstrap.
- Web Apps: To create consistent user interface elements, many web applications make use of Bootstrap's components.
- Mobile-First Approach: Because it is responsive, it is used in the development of mobile-first web apps.

.NET

Microsoft created the adaptable and durable.NET framework, which provides a broad foundation for creating a range of applications, from desktop and web to mobile and cloud-based ones.

Figure 30 : .NET

Advantages and Disadvantages of .NET

Advantages	Disadvantages
• Language Interoperability: It allows developers flexibility by supporting a variety of languages, including C#, VB.NET, and F#.	• Windows-Centric: Prior to the release of.NET Core,.NET was tightly associated with Windows, which restricted its applicability to other operating systems.
• Security Features: Code access security and role-based security are two of the built-in security features of.NET.	• Complexity: Newcomers or those working on small-scale projects may find the framework's vastness overwhelming.
• Large Library: For a variety of functions, it has a large class library called the Framework Class Library (FCL).	• License Fees: Purchasing licenses may be necessary for certain functionality and enterprise-level tools.
• Cross-Platform Development: Applications may operate on many OS environments thanks to.NET Core's cross-platform development capabilities.	• Community Support: The.NET community may be smaller in certain areas than that of other frameworks.
• Performance and Scalability: Just-In-Time compilation makes.NET programs renowned for their performance and scalability.	• Vendor lock-in: Although.NET Core is free and open-source, some Microsoft services and products may lead to vendor lock-in.

Table 10 : Adavantages and Disadvantages of .net

Uses of .NET

- Enterprise apps: Building scalable enterprise-level apps is a common usage for.NET.

- Web Applications: It is used to create web applications, particularly when server-side scripting is done with ASP.NET.

- Desktop apps: WPF and WinForms are two technologies that can be used with.NET to create Windows desktop apps.

- Cloud-based Solutions: Cloud-native applications are made with.NET Core.

- Game Development: To create interactive games, game developers use Unity and.NET.

Angular.js

Google created the front-end JavaScript framework known as Angular.js, or Angular 1. It does this by extending HTML and making reusable components possible, which makes it possible to create dynamic web apps.

Figure 31: Angular.js

Advantages and Disadvantages of Angular.js

Advantages	Disadvantages
• Two-Way Data Binding: Two-way data binding is implemented by Angular.js, which automatically synchronizes updates to models and views.	• Performance: Two-way data binding and digest cycles may cause performance issues as applications expand.
• MVC Architecture: It has a clear, modular code structure and adheres to the Model-View-Controller (MVC) paradigm.	• Steep Learning Curve: Because of its intricate principles and terminologies, beginners may find learning Angular.js difficult.

• Directives: HTML may be made more dynamic and interactive by using the directives that Angular.js offers.	• Maintainability: Because of its versatility, large-scale applications can make code maintenance difficult.
• Dependency Injection: It facilitates the construction of clear, testable code by supporting dependency injection.	• Limitations related to SEO: Angular.js applications may need extra setups.
• Strong community and extensive ecosystem of libraries and extensions are features of this platform.	• Learning Curve: If developers are unfamiliar with front-end frameworks, it may take some time to become proficient in Angular.js.

Table 11 : Adavantages and Disadvantages of angular.js

Uses of Angular.js

- Single Page Applications (SPAs): With its wealth of capabilities and data binding, Angular.js is perfect for creating SPAs.

- Dynamic Web apps: It is employed in the development of dynamic and interactive web apps.

- Prototyping: For rapid prototyping and proof-of-concept development, developers frequently utilize Angular.js.

- Enterprise apps: Because of Angular.js's modular design, a lot of enterprise-level apps make use of it.

- Real-Time Applications: Because of its reactiveness and two-way data binding, it is used in real-time applications.

Libraries

jQuery

jQuery is a well-liked, quick, and clean JavaScript toolkit made to make traversing and modifying HTML documents, event handling, and animation easier.

Figure 32 :Jquery

Advantages and Disadvantages of jquery

Advantages	Disadvantages
• Simplicity: Uses clear syntax and techniques to make difficult jobs simpler.	• Performance Overhead: The speed of jQuery can be impacted by large codebases.
• Cross-Browser Compatibility: Fixes discrepancies between different browser versions.	• Dependency: Developing JavaScript skills may be hampered by an excessive reliance on jQuery.
• DOM Manipulation: Makes it simple to work with HTML and DOM elements.	• File Size: The library's file size may affect load times if it is not optimized.
• Support for AJAX: Makes asynchronous request processing easier.	• Framework Shift: In certain situations, modern frameworks eliminate the necessity for jQuery.
• Flexibility: Provides a vast collection of plugins for further features.	• Learning Curve: Mastering the fundamentals of JavaScript may be limited if one only uses jQuery.

Table 12 : Adavantages and Disadvantages of jquery

Uses of Jquery

- DOM Manipulation: Makes manipulating and traversing HTML documents easier.
- Event Management: Managing clicks, scrolls, and other events with ease.
- Animations: slick transitions and animations for websites.
- Sending asynchronous queries to load dynamic content is known as AJAX requests.
- Plugins: Combining different plugins to provide more features.

Databases

MySQL

MySQL is a relational database management system (RDBMS) that is available for free and makes use of SQL. It powers a lot of data-intensive apps on the web.

Figure 33 : Mysql

Advantages and Disadvantages of Mysql

Advantages	Disadvantages
• Cost-effective: MySQL has a robust community and is available for free as an open-source utility.	• Restricted Functionality: Doesn't have all the functionality that enterprise-level databases do.
• Performance: Well-known for its quickness, particularly in procedures involving a lot of reading.	• Transaction Support: In sophisticated transaction-based systems, it could have restrictions.

• Scalability: Facilitates simple vertical and horizontal scalability to meet expanding needs.	• Analytics Tools: Lacks sophisticated built-in analytics functionality.
• Ease of Use: An intuitive UI and an easy setup procedure.	• Problems with concurrency: High writing loads might cause concurrency issues.
• Strong security features are included for the protection of data.	• Maturity Issues: Regarding important programs, some consumers are unsure of its maturity.

Table 13 : Adavantages and Disadvantages of Mysql

Uses of MySQL

- Web apps: Because of its scalability and speed, web apps are best suited for it.
- E-commerce Sites: Because they are dependable, e-commerce platforms employ them extensively.
- Content Management Systems (CMS): MySQL is used by many CMS, including WordPress, to store data.
- Analytics and Reporting: Applied to basic reporting and analytical duties.
- little to medium-sized Businesses: Because they are affordable, SMBs frequently use them.

Oracle

The relational database management system (RDBMS) known as Oracle Database was created by Oracle Corporation. It is used in businesses and is renowned for its security, scalability, and resilience.

Figure 34 :Oracle

Advantages and Disadvantages of Oracal

Advantages	Disadvantages
• Excellent scalability to manage big volumes of users and data.	• Cost: Generally more costly than alternatives that are open-source.
• Advanced security techniques are employed to safeguard data in security features.	• Complexity: Skill-specific knowledge may be needed for intricate setup and management.
• great Performance: Designed with complex queries and great performance in mind.	• Resource-intensive: May need a significant amount of gear and be resource-intensive.
• Advanced Functionality: Offers a variety of capabilities to meet a wide range of requirements.	• Vendor lock-in: Situations involving proprietary nature may result in vendor lock-in.
• Reliability and Durability: Reliability and data durability are well-known.	• Licensing: Difficult licensing models that may raise the final price.

Table 14 : Adavantages and Disadvantages of Oracal

Uses of Oracal

- Applications for Enterprises: Perfect for big businesses managing massive amounts of data.
- Financial Systems: Because of their security and dependability, financial organizations use them.
- Platforms for e-commerce: Fit for extensive online sales.
- Mission-Critical Systems: Frequently used in setups where high availability is required.
- Government Organizations: Because of its security, it is frequently utilized in government databases.

SQL Server

Microsoft created the relational database management system known as SQL Server. It is renowned for its functionality, performance, and compatibility with other Microsoft products.

Figure 35: SQL Server

Advantages and Disadvantages of SQL Server

Advantages	Disadvantages
• Easy to Use: Simple to set up and maintain, with an intuitive UI.	• Expensive: SQL Server licenses and related expenses might be substantial.
• Integration: Smooth interaction with additional Microsoft services and products.	• Platform Dependency: Limited cross-platform use due to Windows-centric architecture.
• Performance: Outstanding performance, particularly in Windows-based settings.	• Restricted functionality: It may not have all the functionality that some other databases do.
• Scalability: The ability to grow with increasing data and user needs.	• Resource-Intensive: May demand a significant amount of hardware and a lot of resources.
• Security: Provides strong security features and adherence to norms.	• Supplier lock-in: A proprietary system may restrict options for suppliers and flexibility.

Table 15 : Adavantages and Disadvantages of Sql Server

Uses of SQL Server

- Business Applications: Used extensively for a range of purposes in enterprises.
- Fit for both business intelligence and data warehousing applications.
- E-commerce Platforms: Applicable to e-commerce software.
- CMS systems are frequently coupled with content management systems.
- Small and Medium-Sized Businesses: Because of its user-friendliness, SMBs have adopted it

The influence of search engines on website performance and evidence-based support for improving a site's index value and rank through search engine optimization

Search engine

Search engines are complex programs made to search, browse, and index the huge quantity of data that is available online. They act as entrance points to the digital world, giving consumers access to a vast array of information depending on their search terms. These systems search the internet using specialized software called web crawlers, sometimes called spiders, that navigate between pages via links to gather data about websites and their contents. The basis for search engine operation is this procedure, called crawling.

After it has been crawled, the material is indexed, then stored and arranged in databases. This enormous amount of data is analyzed and cataloged by search engines so that users can access it when they conduct searches. In order to rank and present the most relevant results on the search engine results pages (SERPs), the search engine uses sophisticated algorithms to match the keywords entered by the user with the indexed content.

Various ranking parameters are used by Google, Bing, Yahoo!, and other search engines to decide the order in which results are displayed, with the goal of giving users the most accurate and helpful information possible. These search engines are always improving their algorithms to provide people all over the world with better and more relevant results.

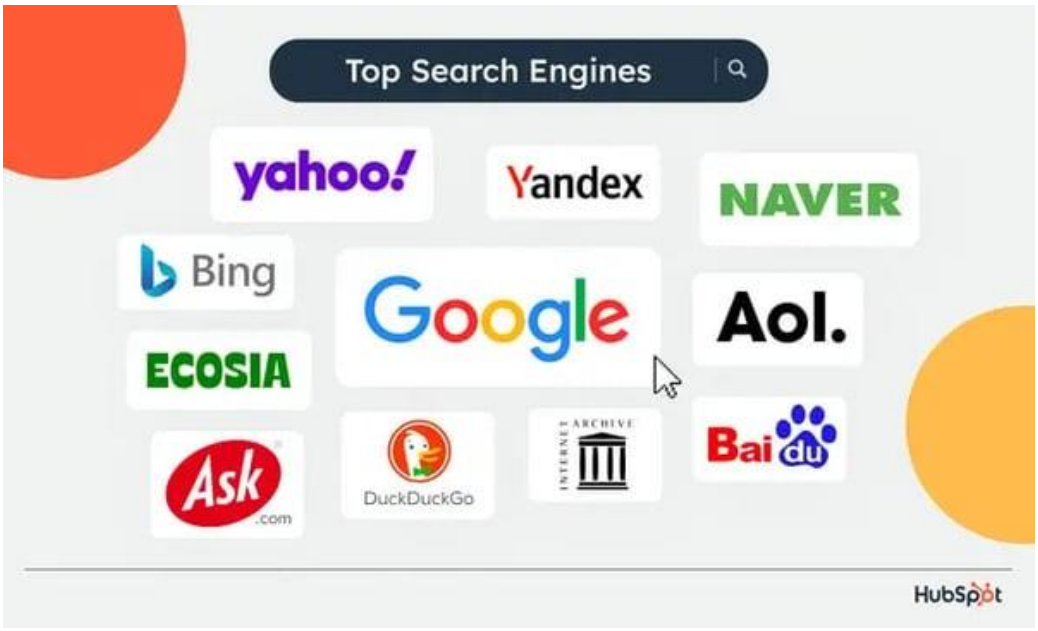


Figure 36 : Top Search Engines

Search Engine Optimization

The goal of search engine optimization (SEO) is to increase a website's visibility and rating on search engine results pages (SERPs) through a variety of tactics and techniques. Here's a summary of the essential elements:

- Crawling: To search the internet and traverse web pages, search engines employ automated bots, sometimes known as crawlers or spiders. These crawlers collect data and content to index by following links from one page to another. For effective crawling, a website's structure must make it easy for these crawlers to navigate.

- Indexing: Following the crawl, search engines compile the gathered data into enormous databases and arrange it. Analyzing and comprehending the relevance and context of the content is necessary for indexing. Websites that are useful and well-structured have a higher chance of being appropriately indexed.

- Ranking: Based on a number of variables, search engines rank web sites using intricate algorithms. Authority, quality, and relevance are essential for ranking. Websites that

use SEO techniques seek to increase traffic and visibility by optimizing elements that affect ranking.

- Keyword research: It's important to comprehend the words and phrases that people type into search engines. Websites can strategically insert high-traffic keywords related to their content into their pages by conducting keyword research.

- On-Page Optimization: This is the process of making certain web pages more search engine friendly. It entails adding pertinent keywords to optimize meta tags, headings, and content while maintaining user value and readability.

- Content Quality: Producing relevant, interesting, and high-quality information is crucial. Content that fulfills user purpose, provides insightful information, and enhances the user experience is given priority by search engines.

- Link Building: One of the most important off-page SEO strategies is obtaining high-quality backlinks from relevant and respectable websites. It gives a website authority and credibility in search engine eyes.

- Mobile responsiveness: Websites must be made mobile-friendly due to the growing use of mobile devices. In mobile search results, mobile-friendly websites appear higher.

- Site Speed: Search engines prefer and offer better user experiences on websites that load more quickly. Enhancing the speed of a website boosts both search engine rankings and user pleasure.

The tools and techniques chosen to realize the Arogya Health Care Website

Arogya Health Care Hospital is a well-known medical facility committed to offering top-notch healthcare services. Known for its patient-first philosophy, it provides all-inclusive medical care. This prestigious facility ensures the finest standards of treatment by fusing state-of-the-art technology with a caring staff of medical professionals.

Arogya Health Care Hospital is in the forefront of providing high-quality healthcare to its community because of its dedication to innovation and holistic therapy.

Current System Challenges

The hospital Arogya Health Care is currently having trouble handling vital information using an antiquated manual system. The use of paper forms, which are dispersed throughout the hospital's infrastructure, results in inaccurate data and departures from management guidelines.

Duplication of data across forms leads to inconsistent data and prevents effective data management. Crucial data including as patient histories, personnel particulars, room reservations, operating room administration, and wait lists are dispersed among multiple paper-based systems.

Proposed Hospital Management System (HMS)

The mission of as the assigned web developer is to design an all-inclusive and reasonably priced Hospital Management System (HMS) for Arogya Health Care. The goal of this system is to simplify and consolidate essential functions:

- Patient Information Control

The process of organizing patient data entails building a single database that contains both medical and personal information. Ensuring thorough and well-organized patient records, this central database improves accessibility while upholding management guidelines.

- Room Availability Management

Creating a strong system to oversee the scheduling of rooms and wards allows for real-time allocation and monitoring, which maximizes the use of resources within the hospital infrastructure.

- Staff and Operating Room Schedules

It's essential to have effective staff scheduling and OT management. Smooth hospital operations are ensured by developing a system that manages operating theater scheduling, personnel shifts, and assignments.

- Patient Invoicing System

Putting in place a single system for patient invoices facilitates effective financial management by reducing insurance information, payment records, and billing procedures.

Author's Choice of Technology Stack

Much consideration was given to choosing a technology stack that could meet every requirement of the healthcare system in the goal to build an effective Hospital Management System (HMS) for Arogya Health Care.

HTML5 and CSS3

When designing the visual components and making sure the system has a responsive interface, HTML5 and CSS3 were the obvious choices. Semantic organization for web content is provided by the versatile HTML5, and the ability to create dynamic and visually appealing layouts is enabled by CSS3. When combined, they guarantee an aesthetically pleasing and intuitive interface that is in line with contemporary design trends, making it simple for medical personnel in all departments to use.

Visual Studio Code

Because to flexibility, security, and large plugin ecosystem, Visual Studio Code was the favored Integrated Development Environment (IDE). Its user-friendly interface and robust editing capabilities speed up development by enabling effective code writing, debugging,

and teamwork. Its cross-platform compatibility guarantees smooth development across many operating systems, increasing efficiency all around.

PHP and MySQL in XAMPP Stack

PHP was selected because of its ability to handle dynamic content through server-side scripting. Its adaptability and database compatibility make it the perfect option for creating a system with a range of data requirements, guaranteeing smooth communication between the HMS's front-end and back-end components. The XAMPP stack's MySQL database management system gained popularity because to its dependability, scalability, and strong security features. It provides effective methods for storing and retrieving data, which is essential for safely handling the large amount of healthcare data.

Conclusion

In conclusion, the carefully chosen technology stack seeks to provide robustness, security, and efficiency in managing the vast hospital data while providing an aesthetically appealing and responsive interface for the HMS of Arogya Health Care. The XAMPP stack's HTML5, CSS3, Visual Studio Code, PHP, and MySQL components offer a unified and expandable solution that satisfies the changing technical needs of the healthcare sector.

The capabilities and relationships between front-end and back-end website technologies and how these relate to presentation and application layers

There is a basic difference between client-side and server-side programs in the context of web development.

Front-end (client-side) technologies include JavaScript, CSS, and HTML. Web content is structured by HTML, styled and designed by CSS, and interactive by JavaScript.

Reverse (Server-side) Technologies consist of Node.js, Ruby on Rails, and PHP. Before transmitting content to the client, they handle data, carry out activities, and create content dynamically on the server.

With the client-side handling user interaction and presentation and the server-side handling data processing and logic execution, this divide guarantees effective and organized web development. Together, the two sides produce a unified and useful online application.

Client-side (Front-end) Technologies

The basis of web content is made up of front-end technologies like HTML (Hypertext Markup Language). Web pages are defined by HTML, which also defines elements like headings, paragraphs, and graphics. It establishes the framework for the functionality of JavaScript and CSS.

HTML elements are presented better when they use CSS (Cascading Style Sheets). CSS guarantees a visually pleasing and uniform appearance across various web pages by specifying styles, layouts, and visual appearances. It is in charge of things like layouts, fonts, colors, and responsive designs.

One especially important reactive scripting language is JavaScript. It gives websites more responsiveness and engagement. JavaScript enables programmers to handle events, generate interactive user interfaces, verify client-side information, and create requiring user experiences.

Server-side (Back-end) Technologies

PHP is a scripting language that runs on servers and is mostly used for web development. By producing material on the server before transmitting it to the client, it makes it possible to create dynamic web pages. PHP allows for the smooth integration of databases and HTML, making it possible to perform sophisticated functions like data processing, session management, and user authentication.

The powerful web development framework Ruby on Rails is based on the Ruby programming language. It provides a convention-over-configuration method that emphasizes code productivity and simplicity while expediting development. It is known for its toughness, effective code reuse, and quick prototyping.

A runtime environment called Node.js is used to run JavaScript scripts on the server side. It is praised for having an event-driven design, which allows for asynchronous I/O operations and is perfect for creating real-time, scalable applications. Because Node.js uses a single-threaded, non-blocking paradigm, it operates with great efficiency and performance.

The front-end and back-end communication

Every interactive web application is built on the basis of front-end and back-end communication. Users quickly interact with the HTML, CSS, and JavaScript that make up the front-end. The front-end communicates with the back-end, which is made up of databases and server-side languages, when a user engages with the interface, for as by completing a form or requesting data.

The usual method of this engagement is through HTTP requests. For example, a POST request containing the user's inputs is sent to the server when a user submits a form. This request is handled by the back end, which also manages business logic, queries databases, and carries out any computations that are required. After then, the back-end produces a response, which is frequently produced as HTML, JSON, or XML.

After that, the response is delivered back to the front end, where it fulfills the necessary action or gives the user the information they requested. Users will receive accurate data and responses depending on their inputs or actions thanks to this smooth interaction, which enables dynamic and real-time updates on the front end.

This communication guarantees the smooth operation of web applications, providing a fluid and interactive user experience, whether it is synchronous, meaning the front-end waits for a response before proceeding, or asynchronous, meaning the front-end can move forward with other duties while it waits for a delayed response.

The way website technologies are affect on presentation and application layers.

Application layer

The application layer, which resides on the server-side, includes databases like MySQL, MongoDB, and PostgreSQL as well as a variety of technologies like PHP, Python, Ruby, and Node.js. This layer handles data processing, retrieval, and storage in the background. Back-end technologies manage server functions, including traffic routing, request processing, and strong security mechanisms that protect private user information. Here, authorization and authentication systems play a crucial role in granting users access while preserving privacy. The back-end's APIs and endpoints allow the server and client-side to communicate and exchange data.

Presentation layer

On the other hand, user interaction takes place directly on the presentation layer, also referred to as the front-end. It consists of HTML, CSS, and JavaScript, which are in charge of organizing content on websites, styling elements, and promoting interaction, respectively. User experience (UX) is heavily influenced by front-end technologies, which include responsiveness, accessibility, and design features. JavaScript frameworks that provide dynamic functionalities to the front end, such as React or Vue.js, enable a compelling user interface and experience. Through APIs or endpoints, front-end systems can communicate with the back end and retrieve and display data from the server.

Connection between the Layers

In order to create a unified and useful online application, front-end and back-end connection is essential. They depend on smooth cooperation and synchronization to make sure that modifications or updates in one layer are reflected correctly in the other. Both layer performance enhancement improves overall efficiency and speeds up load times. Furthermore, scalable and adaptive architectures at both layers enable applications to develop and expand effectively without requiring significant reconfigurations. The functionality, performance, and user experience of web applications are powered by the mutually beneficial interaction between the front-end and back-end.

The differences between online website creation tools and custom built sites with regards to design flexibility, performance, functionality, User Experience (UX) and User Interface (UI).

Online Website Creation Tools

Tools for creating websites on the internet have completely changed how people and companies have an online presence. Because of their user-friendly interfaces, these platforms do not require deep technical knowledge or coding experience. With the help of configurable templates, drag-and-drop functionality, and other capabilities, users can quickly and effectively construct websites with a polished appearance.

These technologies serve a range of purposes, from e-commerce stores and corporate websites to personal blogs and portfolios. They frequently provide a range of pre-designed templates, giving customers the option to choose ones that complement their brand's aesthetics or particular needs. The usefulness and visibility of the website are further improved by the built-in features that many of these platforms provide, such as mobile friendliness, SEO optimization, and integration with third-party applications.

Wix, Squarespace, Weebly, WordPress.com, and Shopify are a few well-known online tools for creating websites; they all have distinct capabilities and cater to diverse user bases. These tools enable a wide spectrum of people who want to have an online presence to develop visually beautiful, functional, and flexible websites without requiring substantial technical knowledge.

- Wix

Wix is known for its simple drag-and-drop user interface, which makes it suitable for both novice and expert users. It provides a wide range of aesthetically pleasing themes for different businesses, so even without coding experience, creating beautiful websites is simple. With the help of the platform's App Market, users can incorporate tools for marketing, social media, e-commerce, and form integration. Although the free plan is accessible, it has limited capabilities and Wix-branded advertisements. Advanced features like more storage, unique domain names, and e-commerce potential are only available with

premium accounts. Wix is a great option for small enterprises, portfolios, and creatives looking to create visually striking websites because it places a high priority on design freedom.

- Squarespace

Squarespace is well-known for its intuitive user interface and elegant design aesthetics. It provides a selection of cutting-edge, adaptable templates ideal for a range of industries, with a focus on eye-catching layouts and styles. With the platform's integrations for analytics and third-party applications, customers may easily customize styles and content. Squarespace guarantees a smooth website construction experience by offering dependable hosting, security, and customer support. Though it's appropriate for small to medium-sized enterprises, its e-commerce features might not be as robust as those of specialized e-commerce platforms.

- Weebly

Weebly is a popular option among novices due to its user-friendly interface and basic features. Its extensive template library and drag-and-drop editor make creating websites easier. Weebly offers e-commerce capabilities, blogging features, and basic SEO tools to meet a range of demands. Although Weebly's customization choices may be somewhat more limited than those of other platforms, its easy-to-use interface and reasonable price make it a viable option for individuals who want to rapidly construct functioning websites without requiring complex customisation.

- WordPress.com

Known for its scalability and versatility, WordPress.com is a popular and flexible platform. It has a number of features that are appropriate for huge corporations, small businesses, and blogs alike. The platform offers a range of customisable themes and layouts to meet different demands, from intricate e-commerce sites to basic blogs. Users can efficiently generate and manage content because to its user-friendly editor and straightforward UI. Although there is a free plan accessible, paying plans come with more customization features including premium themes, plugins, and custom domains. WordPress.com is a

great option for content-driven websites because of its powerful blogging functionality and SEO-friendly features.

- Shopify

Shopify is a specialized e-commerce platform made for companies who want to set up and run successful online stores. It offers a one-stop shop for building, modifying, and maintaining e-commerce websites. A vast selection of expertly created themes and templates made especially for online stores are available on Shopify. It has integrated capabilities for order processing, payment gateways, shipping, and inventory management. Through a variety of apps and connectors, the platform's app store enables users to increase functionality and offers choices for marketing, customer service, and analytics. Shopify may not be as adaptable for non-commerce websites as WordPress.com, even though its primary concentration is e-commerce. However, it is a top option for companies wishing to have an online presence due to its user-friendly layout and strong e-commerce functionality.

Custome Built sites

- Design Flexibility

Unmatched design versatility is one of the main benefits of having a website custom-built. Developers are free to create one-of-a-kind, customized designs that perfectly reflect the objectives and identity of the brand. Custom sites, in contrast to templates, are unrestricted by preset frameworks, enabling creative layouts, interactive features, and unique aesthetics.

- Performance Optimization

Websites that are specifically designed can have their performance enhanced. Faster load times and a more seamless user experience can be achieved by developers by streamlining code, optimizing pictures, and implementing effective backend architecture. Improved search engine results and increased user retention are two benefits of this strategy.

- Enhanced Functionality

Another important advantage is that the functionality may be customized to match certain needs. Advanced features, unique functionalities, and specialized tools that are tailored to

the demands of the business or users can all be integrated into custom websites. Custom-built websites can handle a wide range of functionalities, including intricate databases, interactive features, and distinctive user interfaces.

- User Interface (UI) and User Experience (UX)

Custom-built websites put the user experience first by emphasizing responsiveness, accessibility, and easy navigation across a range of devices. User happiness and website interaction can be positively impacted by carefully crafting user interfaces that provide a smooth and captivating user experience.

- Content Management Systems (CMS)

Specifically designed content management systems can be utilized by custom websites. Although flexible platforms like as WordPress or Drupal are available, a custom content management system (CMS) provides more exact control over processes, security protocols, and content categorization. This degree of personalization guarantees effective content management in line with corporate objectives.

- Conclusion

In the end, customized websites offer a special combination of expanded functionality, performance optimization, design freedom, and user-centric interfaces. Whether in design, functionality, or content management, the customized approach to development enables companies to build a digital presence that accurately captures their brand and successfully engages their target market.

The differences between online website creation and custom built website

Online website creation tools, such as Wix, Squarespace, and Weebly, present an accessible entry point for individuals and small businesses to establish an online presence. These platforms provide intuitive interfaces and ready-made templates, enabling users without extensive technical expertise to design and launch websites quickly. While they offer cost-effective solutions with low initial investment and easy maintenance, their design limitations and reliance on platform-specific features may hinder the realization of unique branding or complex functionalities. Conversely, custom-built websites, crafted by skilled

developers or agencies, offer unparalleled design flexibility, tailored functionalities, and scalability. They empower businesses to create unique, high-performance sites aligned precisely with brand identities. However, the increased development time, higher costs, and ongoing maintenance requirements can pose challenges for smaller ventures. Choosing between these options hinges on factors like budget, customization needs, long-term objectives, and the desire for full control and scalability in the online presence.

A range of tools and techniques available to design and develop a custom built website

Front End Frameworks

- Bootstrap

A popular and powerful front-end framework, Bootstrap offers an extensive library of pre-made elements, including forms, buttons, navigation bars, and responsive grid layouts. Bootstrap, created by Twitter, simplifies web development by providing a mobile-first and responsive methodology. It's well-known for being straightforward, accessible to novices, and consistent across a range of devices. If not heavily altered, its popularity could lead to a relatively generic look. Rapid prototype and production-ready applications are made easy with Bootstrap's abundance of themes and plugins, significant community support, and comprehensive documentation.

- Foundation

Focusing on flexibility and modularity, Foundation is a sophisticated front-end framework. It offers a wide range of responsive components and a more adjustable grid system, enabling developers to customize their designs to meet the demands of particular projects. With this framework, semantic HTML is given priority, allowing developers to create webpages that are clear, navigable, and optimized for search engines. Because Foundation is modular, users can include just the components that are actually needed, which speeds up loading times. Compared to Bootstrap, it may have a higher learning curve yet offering greater flexibility. On the other hand, its versatility and thorough documentation make it appropriate for developing original and expandable ideas.

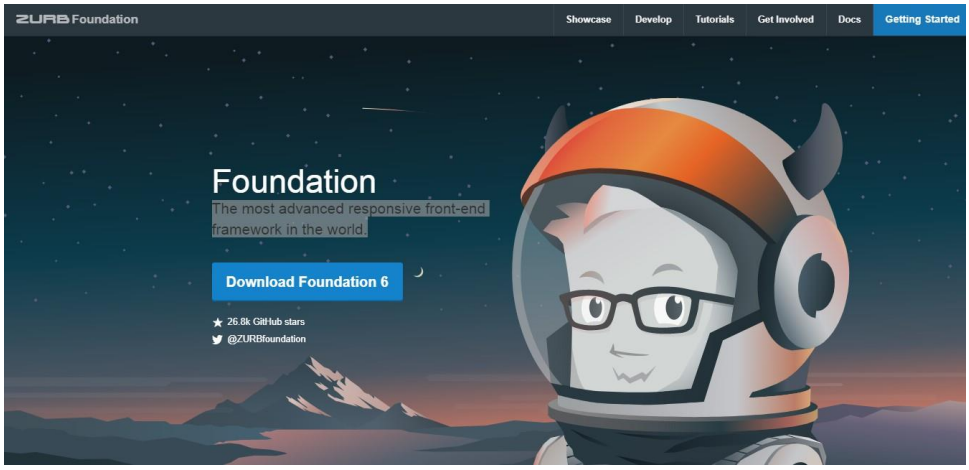


Figure 37 : Foundation

- Semantic UI

Semantic UI uses natural language naming conventions to provide syntax that is easy to understand and understandable by humans. The goal of this framework is to offer responsive, intuitive, and simple-to-learn components. Semantic UI encourages readability and consistency in design, which makes it easier for developers to create aesthetically pleasing interfaces and rapidly understand its functionality. In contrast to Foundation or Bootstrap, it can have a smaller community and fewer resources, which could affect the availability of third-party themes or plugins. Nevertheless, it's a great option for projects where readability and speed of development are critical due to its simple syntax and ease of usage.

Figure 38 : Semantic UI

Web Application Frameworks

- Ruby on Rails

The Ruby programming language serves as the foundation for the sophisticated Rails web application framework. It makes development more effective and less repetitive by adhering to the convention over configuration (CoC) and don't repeat yourself (DRY) concepts. Web application development may be done quickly with Rails because to its extensive ecosystem of frameworks and tools. As a result of its emphasis on convention-based setups and abundance of built-in capabilities for jobs like database management, it is a top pick for new businesses and ambitious projects.

- AngularJS

Google developed and maintains AngularJS, a potent front-end framework built with JavaScript. Its purpose is to create single-page applications (SPAs) that are dynamic. The Model-View-Controller (MVC) architecture is used by AngularJS to make it easier to create sophisticated online apps. By providing two-way data binding, it makes it possible for the model and view components to automatically synchronize. Although the learning curve can be challenging for novices due to its widespread use, developers can find plenty of support from its vast documentation and vibrant community.

- Ember.js

This JavaScript-based framework is also intended for the development of ambitious online apps. It provides a clear framework and prioritizes convention over configuration, which highlights developer productivity. Because of Ember.js's strong views on application structure, developers are able to create scalable and maintainable apps. Ember CLI, routing, and data management are just a few of the many features and tools that come with it.

- Express.js

Express.js is a flexible and simple Node.js web application framework that makes creating web applications and APIs easier. Because of its small weight and neutrality, developers can exert more influence over the architectural design of the program. With the extensive middleware offered by Express.js, responsibilities such as request processing, routing, and

database integration become easier to handle. Because of its ease of use and wide community support, it may be used to develop scalable and light-weight online applications.

- Meteor

Real-time online application development and quick prototyping are made possible by the full-stack JavaScript framework Meteor. It allows for a single codebase to be used for both front-end and back-end development. Among Meteor's advantages are its integrated database management packages and reactivity, which enables real-time adjustments to the user interface without human interaction. Applications needing fast development cycles and real-time data synchronization should use Meteor.

Figure 39: Metor

- Django

Django is a high-level web framework built on Python that is renowned for its ease of use, readability, and adaptability. With a wide range of integrated features like an admin interface, authentication, and an ORM (Object-Relational Mapping) system, it adheres to the "batteries-included" idea. Django is an excellent choice for developing intricate web applications, such as content management systems and e-commerce platforms, because to its stability, scalability, and emphasis on security.

Figure 40 : Dajango

- ASP.NET

Microsoft created the ASP.NET web application framework, which is mostly used to create dynamic web pages and services. It is compatible with several programming languages, including as F#, Visual Basic, and C#. ASP.NET offers a range of tools and frameworks to facilitate the development of web applications. These include ASP.NET Web API for constructing RESTful services and ASP.NET MVC for building web applications with the Model-View-Controller paradigm. It offers strong enterprise application support and works seamlessly with other Microsoft technologies.

Languages / Platforms

- PHP: PHP is a scripting language that runs on servers and is mostly used for web development. It is frequently used to construct dynamic web pages and web apps and is incorporated into HTML. PHP is a well-liked option for developing web solutions because of its adaptability, simplicity of use, and extensive community support. It's frequently used to build reliable backend systems using databases like MySQL.

- Node.js: Node.js is a server-side runtime environment that enables JavaScript programming. With an event-driven, non-blocking I/O strategy that makes it lightweight and effective for real-time applications, it is based on the V8 JavaScript

engine found in Chrome. Building scalable network applications, streaming applications, and APIs is a frequent use case for Node.js.

- JavaScript: Mostly used for front-end web development, JavaScript is a flexible computer language. Its duties include providing dynamic content, enhancing website engagement, and producing user experiences that are responsive. Since the release of Node.js, JavaScript is also frequently utilized for server-side development.

- HTML5 is the most recent version of the Hypertext Markup Language, which is used on the World Wide Web to organize and display content. Multimedia components, enhanced forms, a drawing canvas, and improved APIs for offline web apps are just a few of the new features and improvements it brings. In the creation of responsive and multimedia-rich online apps, HTML5 is particularly important in today's web development.

- Python: Python is an interpreted, high-level programming language that is renowned for being easy to learn and comprehend. It is extensively utilized in data analysis, automation, scientific computing, web development, and artificial intelligence. Python is a popular choice for quick development and experimentation because of its huge library and clear syntax.

- Ruby: Ruby is an object-oriented, dynamic programming language with a focus on productivity and simplicity. It is renowned for its sophisticated syntax and developer-friendly methodology. Ruby is frequently linked to the Ruby on Rails framework, which emphasizes convention over configuration and quick development when developing web applications.

Website Speed test Tools

- WebPageTest: Offering sophisticated testing capabilities, WebPageTest is an open-source program. Users can conduct tests with different browsers and from different places. This application provides comprehensive reports on content breakdown, load times, and improvement recommendations. It is well-known for its filmstrip views, which display loading procedures and visuals, and waterfalls.

Figure 41 : Web page Test

- Pingdom: Pingdom is an easy-to-use application that provides page performance testing and website monitoring. It offers information on performance ratings, load times, and optimization recommendations. Real-time monitoring and alerts when site performance declines are also provided by Pingdom.

Figure 42 : pingdom

- GTmetrix: This all-inclusive tool integrates information from YSlow and Google PageSpeed Insights. It offers a thorough evaluation of the functionality of websites, complete with scores, actionable recommendations, and page load times. Users can test using several browsers and server locations using GTmetrix.

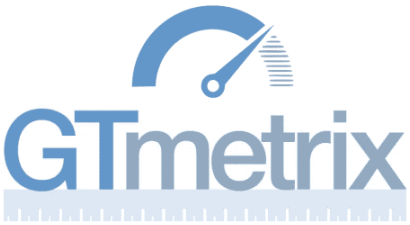


Figure 43 : GTmetrix

- Website Test: Powered by KeyCDN, Website Test provides a streamlined interface for performance analysis of websites. It offers information on waterfall charts, content breakdowns, and load times. Users can test with different browsers and regions, and they will get advice that they can implement.

- DareBoost: DareBoost provides performance reports and in-depth analysis along with practical suggestions. It offers information on SEO, performance ratings, load times, and best practices. In addition, DareBoost provides alerts for variations in performance and ongoing monitoring.

Figure 44: Dareboost

- Google Chrome DevTools: The Chrome browser's built-in web developer tools are called Google Chrome DevTools. It provides a number of capabilities, such as timeline records, performance audits, and network analysis. Website speed bottlenecks can be found and optimized by developers right within the browser.

Figure 45 : Google chrome Devtools

Website Technologies ,tools and techniques with good design principles to create a mutiple website

The manual system that Arogya Health Care now uses to manage vital information presents issues. Incomplete or inconsistently stored data is the result of the hospital's struggle with multiple paper forms dispersed throughout its infrastructure. Multiple sites house different critical documents such as staff details, patient information, and room scheduling, which can lead to inefficiencies and possible data errors.

An all-inclusive web-based Hospital Management System (HMS) is essential to helping with these problems. By centralizing and optimizing information management, this system seeks to transform Arogya's business processes. The first step involves strong user authentication and permission procedures that guarantee safe access for various staff members, including administrators, medical personnel, and receptionists.This HMS's primary features include a wide range of hospital operations activities. Leading the way is patient management, which enables quick registration, medical history recording, and safe electronic health record storage. Furthermore, streamlined scheduling for operations, consultations, and other procedures is made possible by effective room and facility management features that provide real-time visibility into room availability.

Staff management is given top priority by the system, which makes it easier to arrange employee data and schedules to maximize staffing numbers and rotations. While a reception module manages questions and patient movements throughout the hospital, billing and invoicing modules produce precise invoices for services provided.By offering insights into patient demographics, occupancy rates, and revenue patterns, a comprehensive reporting and analytics suite facilitates data-driven decision-making. The solution prioritizes compliance with security standards like HIPAA to secure sensitive patient information and effortlessly interacts with current hospital systems to provide scalability and future-proofing.Such a system must be developed by combining strong back-end languages and databases with front-end technologies like HTML, CSS, and JavaScript frameworks. The HMS that Arogya Health Care envisions will open the door to a more effective, safe, and patient-focused healthcare ecosystem by fulfilling the listed requirements.

Wieframes

- Login

Figure 46 : Login Wireframe

The login page wireframe sets the initial interaction tone for users accessing the Hospital Management System. With a clean and intuitive design, it presents a secure gateway to the system, featuring fields for username and password entry. The wireframe emphasizes simplicity and functionality, ensuring a smooth authentication process for different user roles, such as administrators, doctors, nurses, and receptionists. Clear call-to-action buttons prompt users to log in, fostering a seamless transition to the system's dashboard upon successful authentication.

- Dashboard

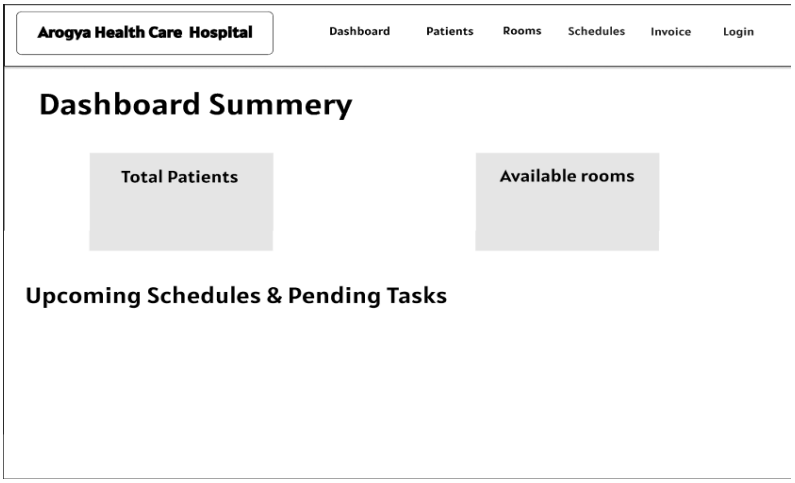


Figure 47 : Dashboard Wireframe

The dashboard wireframe serves as the central hub of the Hospital Management System, offering a comprehensive overview of critical information at a glance. Designed with a user-centric approach, it presents key modules and functionalities in a visually appealing manner. The wireframe incorporates widgets or panels displaying summaries of patient statistics, room availability, pending tasks, and upcoming appointments. Intuitive navigation elements enable swift access to different sections of the system, promoting efficient workflow management for healthcare professionals.

- Patients Registrartiom

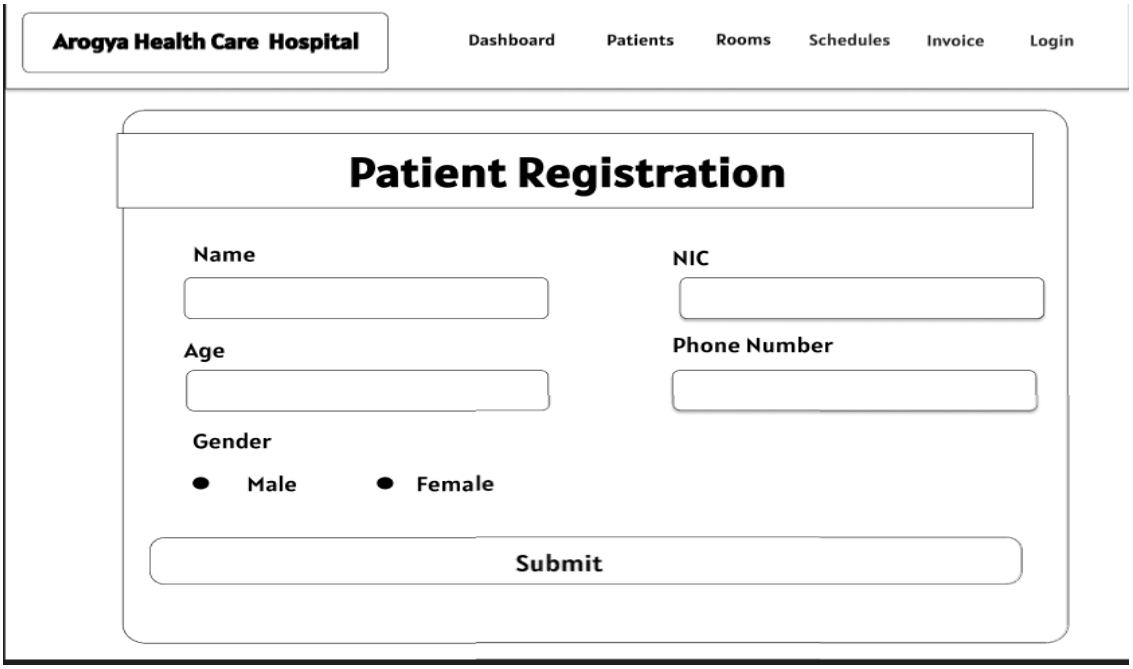


Figure 48 : Patient Registrartiom Wireframe

The patient registration wireframe revolutionizes the process of capturing and storing patient information in the Hospital Management System. Designed with simplicity and thoroughness in mind, it offers a structured interface for registering new patients. The wireframe includes fields to input comprehensive details such as personal information, contact details, medical history, insurance data, and unique patient identifiers. It ensures a seamless registration process while adhering to privacy and data security standards, empowering healthcare professionals to efficiently manage patient records.

- Rooms

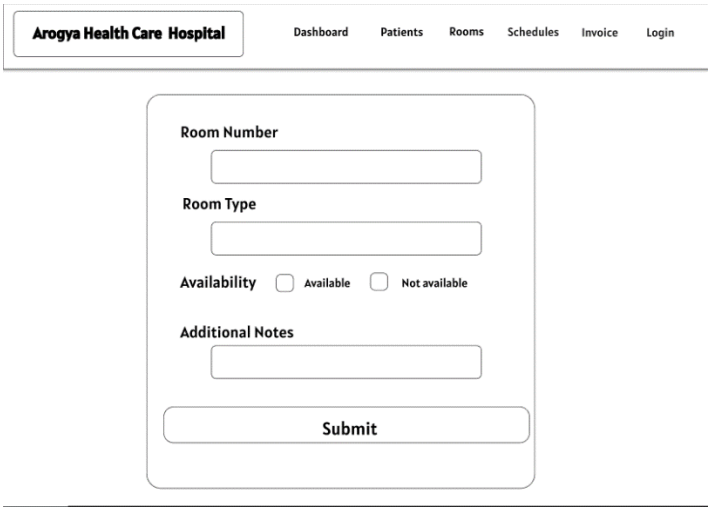


Figure 49 : Rooms Wireframe

The room management wireframe centralizes the oversight of available rooms, wards, and facilities within the hospital premises. It provides a visual representation of room occupancy, availability, and allocation status. The wireframe displays a floor plan or a list-based interface showcasing various rooms and their current statuses (occupied, available, reserved). Users can interact with the wireframe to assign rooms for patient admissions, surgeries, consultations, and other medical procedures. Additionally, it might include features for filtering rooms by type, capacity, or availability, enhancing the ease of room allocation and management.

- Schedules

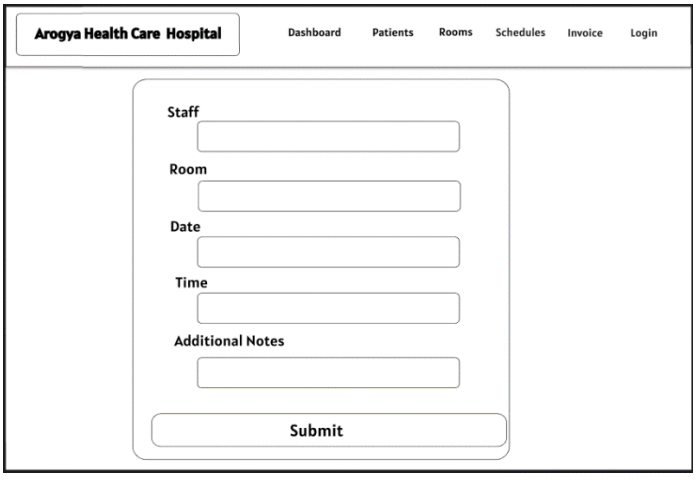


Figure 50: Schedules Wieframe

The scheduling wireframe streamlines the intricate task of managing staff shifts, patient appointments, and facility bookings. It offers a comprehensive view of the hospital's schedule, allowing authorized users to efficiently allocate resources and optimize operations. The wireframe presents a calendar-based interface with clear navigation options and intuitive controls. Users can easily view, create, edit, and delete appointments or shifts, ensuring a well-organized and optimized workflow. Additionally, it may incorporate color-coded categories or filters for different types of schedules (staff, operating theaters, consultations) for better clarity.

- Invoice Login

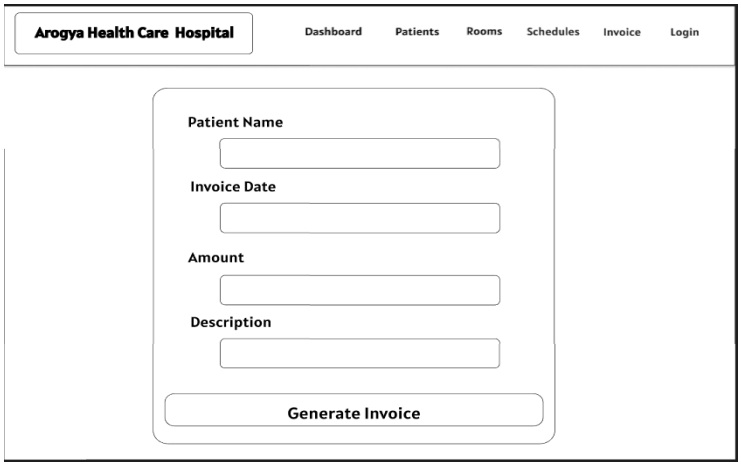


Figure 51 : Invoice wireframe

The invoice management wireframe streamlines the process of generating and managing patient invoices within the Hospital Management System. Through a user-friendly interface, it facilitates the creation of detailed invoices for various medical services provided to patients. The wireframe includes fields for inputting service details, costs, and patient information, ensuring accuracy and transparency in billing procedures. Additionally, it might incorporate features for tracking payment statuses and generating reports to monitor financial transactions efficiently.

Interfaces

- Login

Figure 52 : Login

In this page user can log into the Arogra Health care website by enter the user name ,password and then should select the role of the user. There 3 type roles Admin ,staff, Receptionist. After the click the Login Button.

- Dashboard

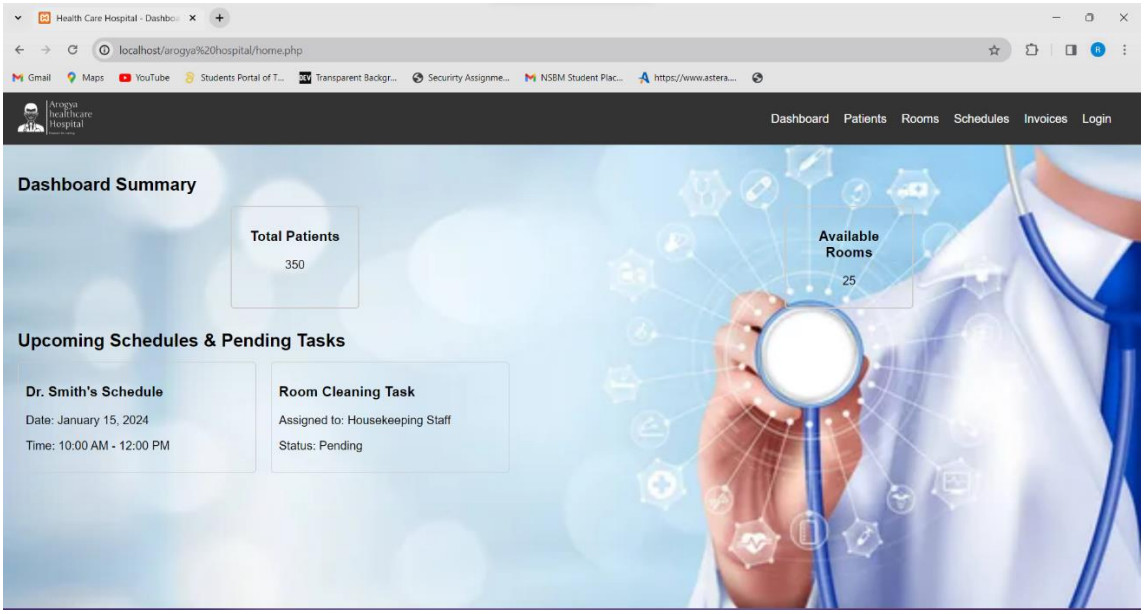


Figure 53 : Dashboard

In dashboard user can looke the dashboard summary about the total patients and also the room that available currently.Also user can look about the upcoming schedules and pending tasks .

- Patients Registrartiom

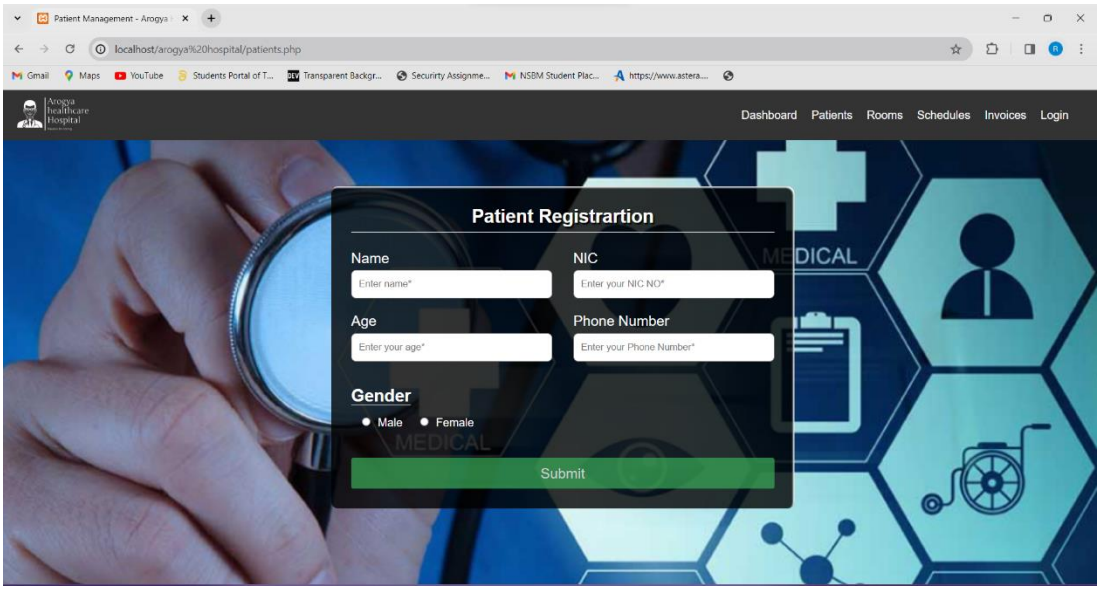


Figure 54 : Patient Registrartiom

In this page user can register.To register a patient need to enter name then NIC number of patients then Age of the patient then Phone number after that should select the gender of the patient then user can click the submit button.

- Rooms

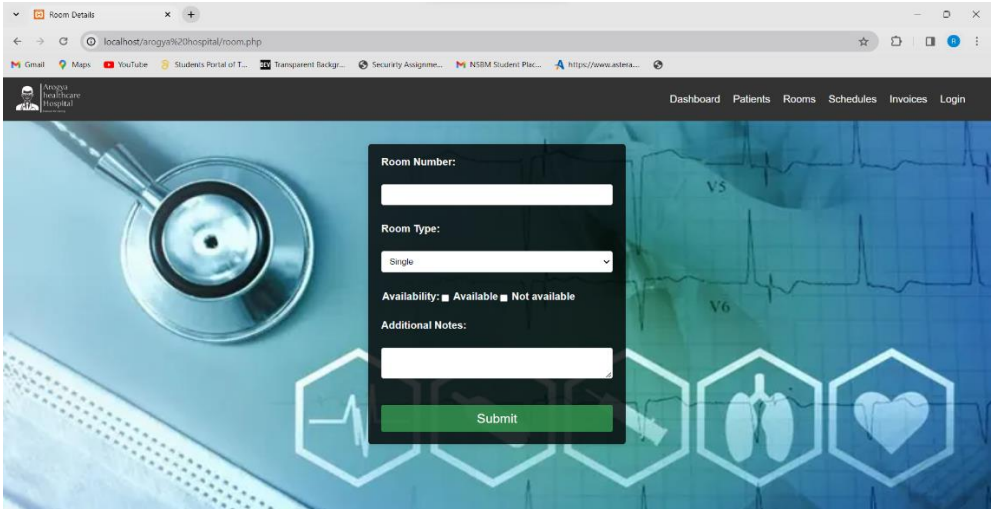


Figure 55 : Rooms

In this page user can look if the room is available or not.So user can enter a room number then a room type. In room type there are 3 categories. So the three categories are Single , Double and Suite. After that user should enter available or not then if need to additional notes user can enter after that click the submit button.

- Schedules

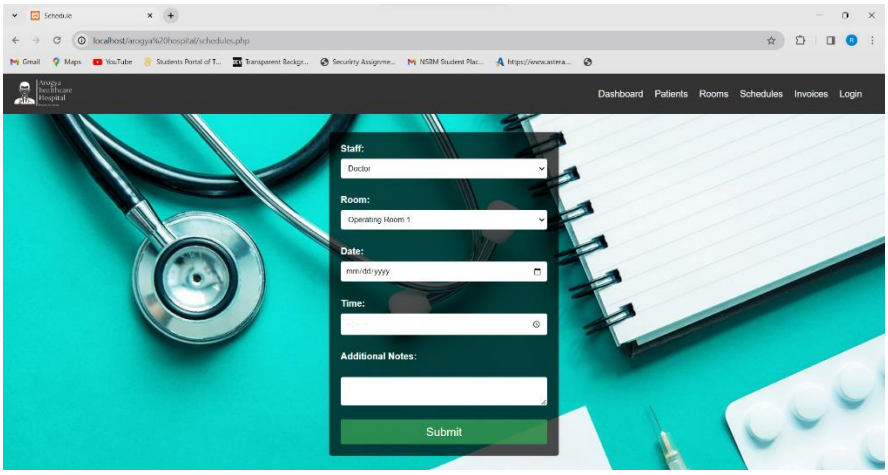


Figure 56: Schedules

In here user need to select from staff there are 3 categories.The types are Doctor,Nurse and Technician . Then need to select Room the author decided to add operating 1 , operating 2 and operating 3 after that can select the date then the time after that if need user can also add additional notes then lastly can click the submit button.

- Invoice Login

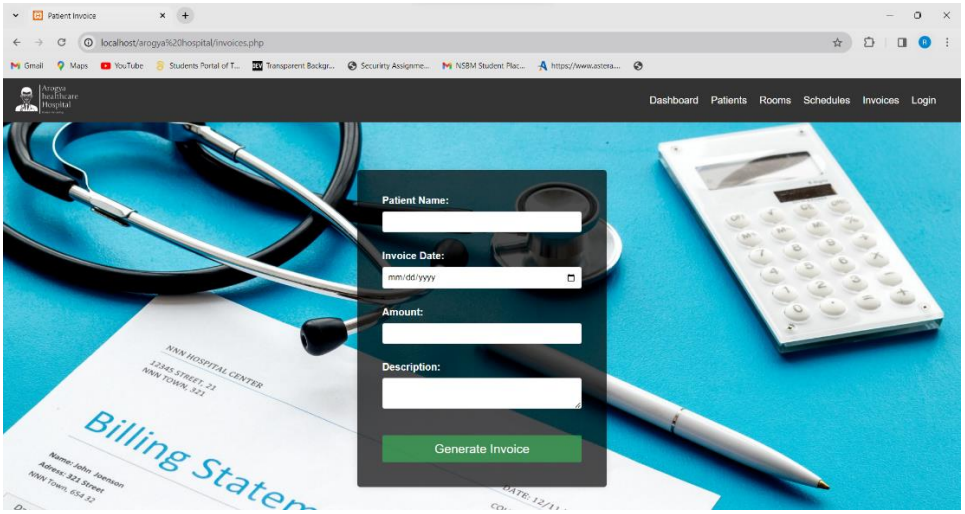


Figure 57 : Invoice

In this page user can add the patients name after that can select the invoice date then the patients charge or amount and after that description then after that can clickthe generate Invoice.

If the user wants to edit the details and also to view the data below are the Recordes of the Tables.

Recorded Tables of the Interfaces

All the records are available in here when user enter the data if user want to edit them user can go to this tables and edit them .In the recordings page display the data and insert ,delete and update buttons. By using that buttons simply user can edit them.

- Patients Record Tables

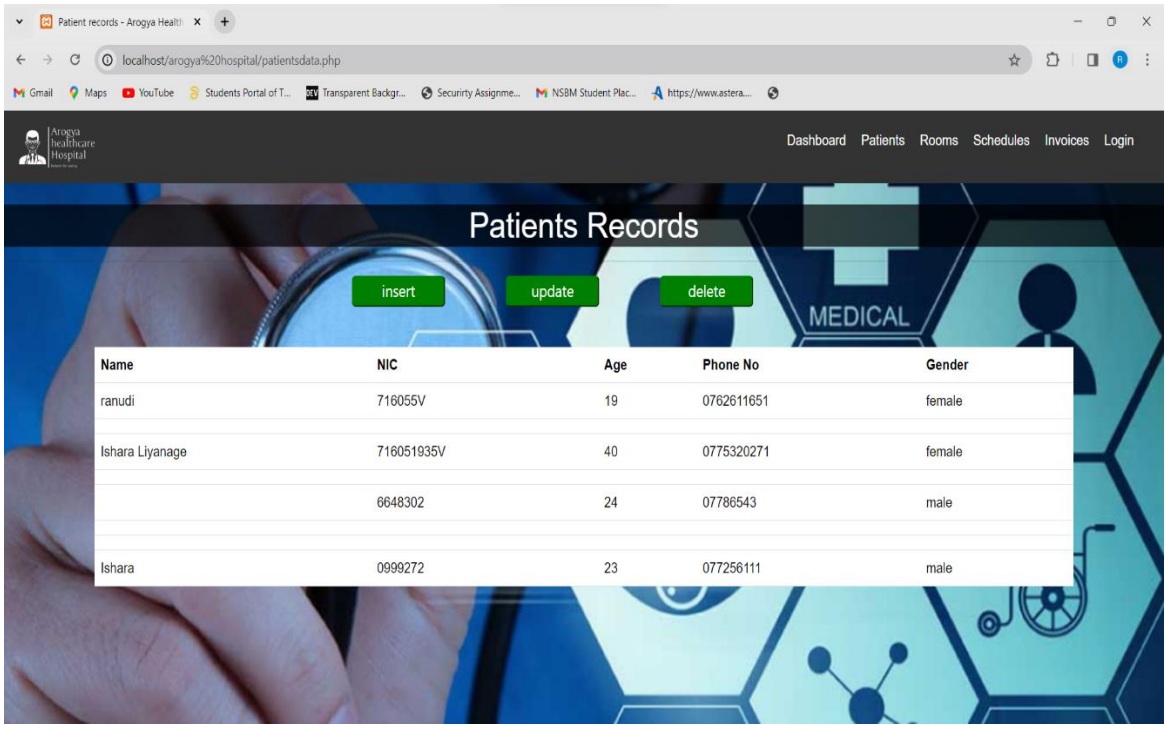


Figure 58 : Patients Records

- Room records Tables

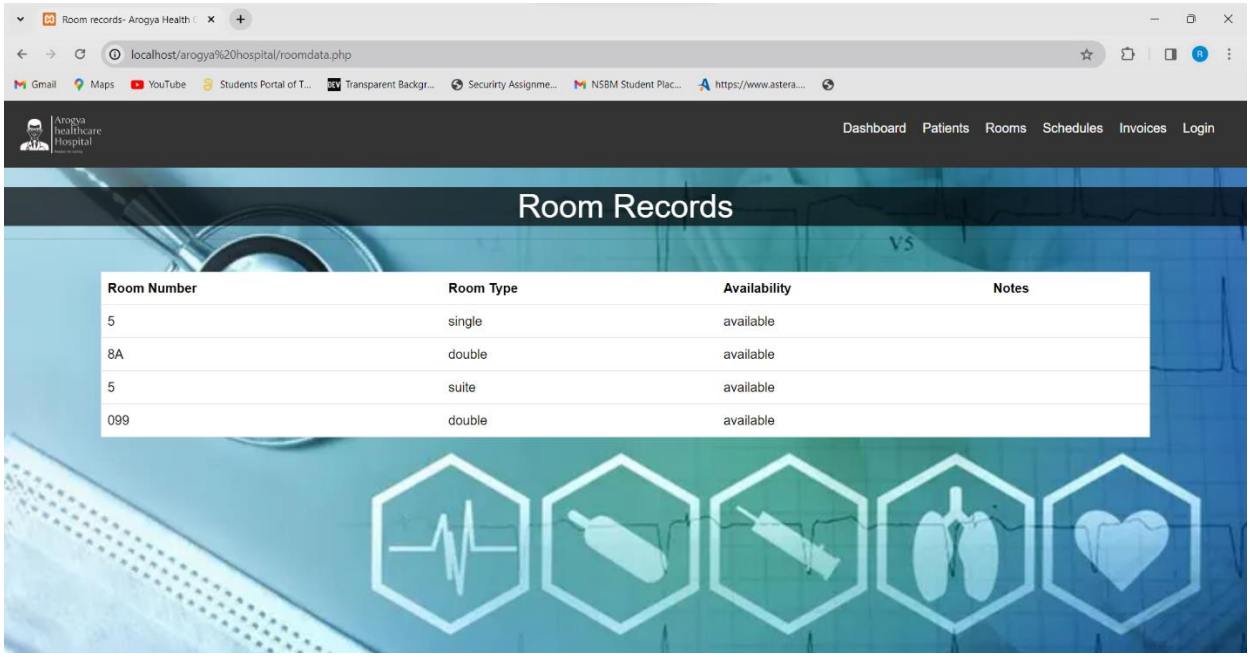


Figure 59 : Room records

- Schedules records Tables

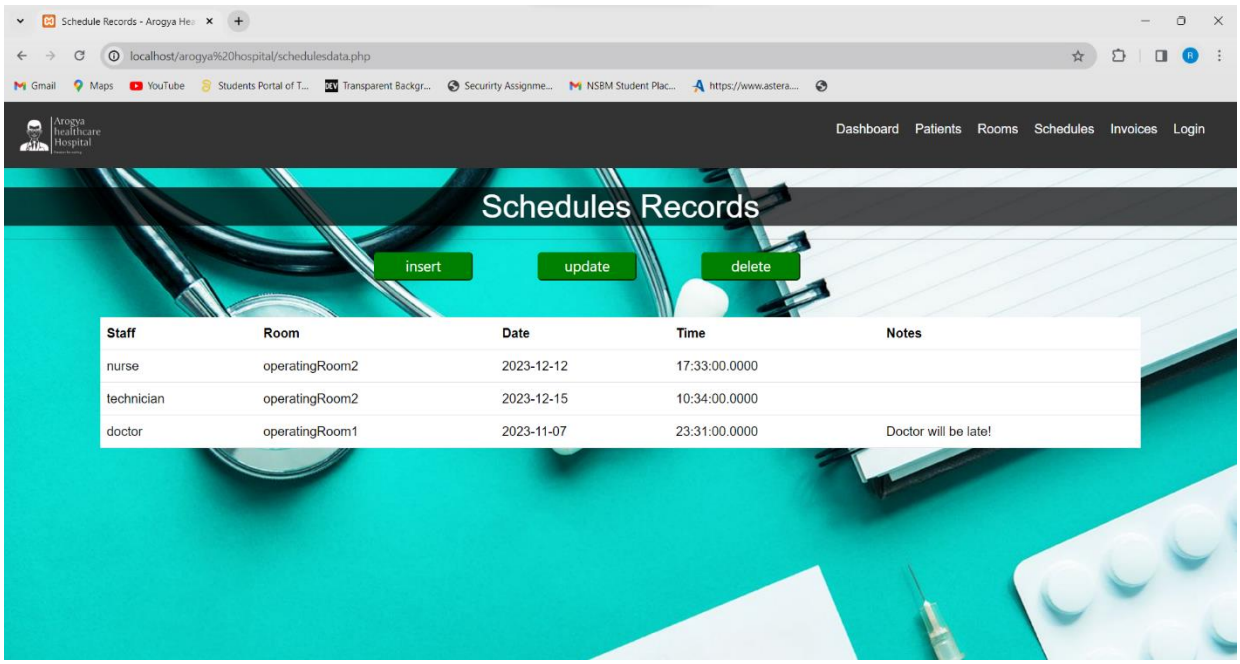


Figure 60 : Schedule Records

- Invoices records Tables

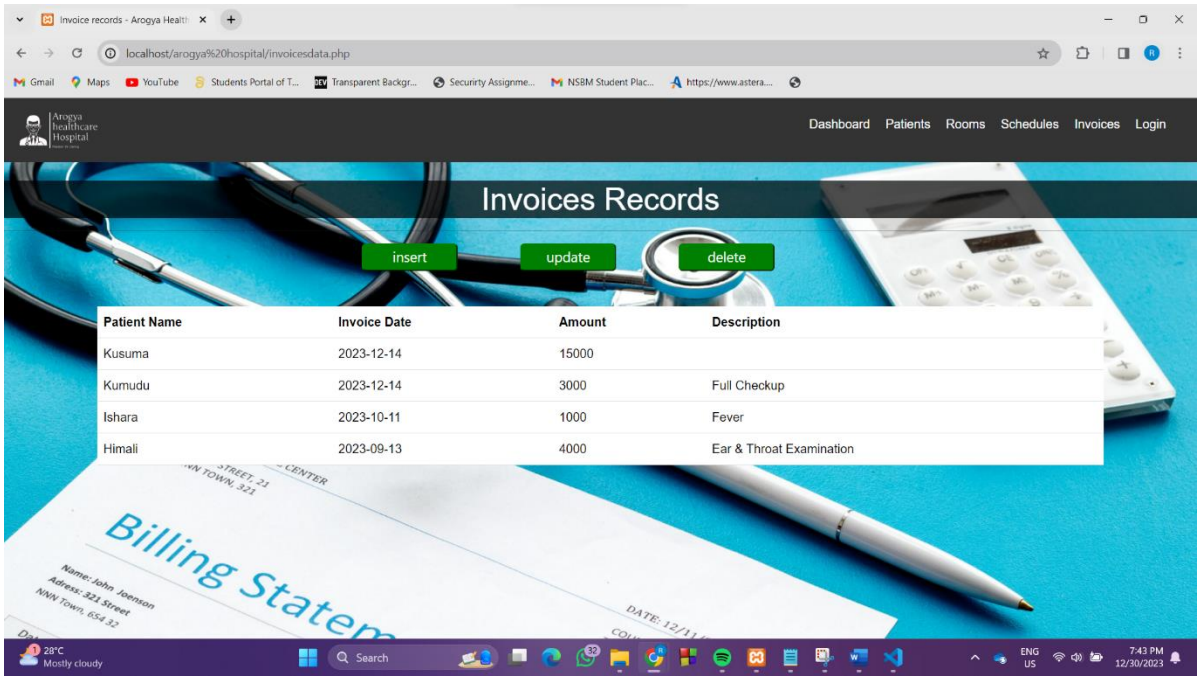


Figure 61 : Invoice records

Writers Codes

Writer has code lots of codes using visual studio code. These are the main codes writer has used in HTML,CSS, PHP and Javascript. So author shows two parts one is interface part and other one is recorded table parts.

Recorded Table Codes

- HTML

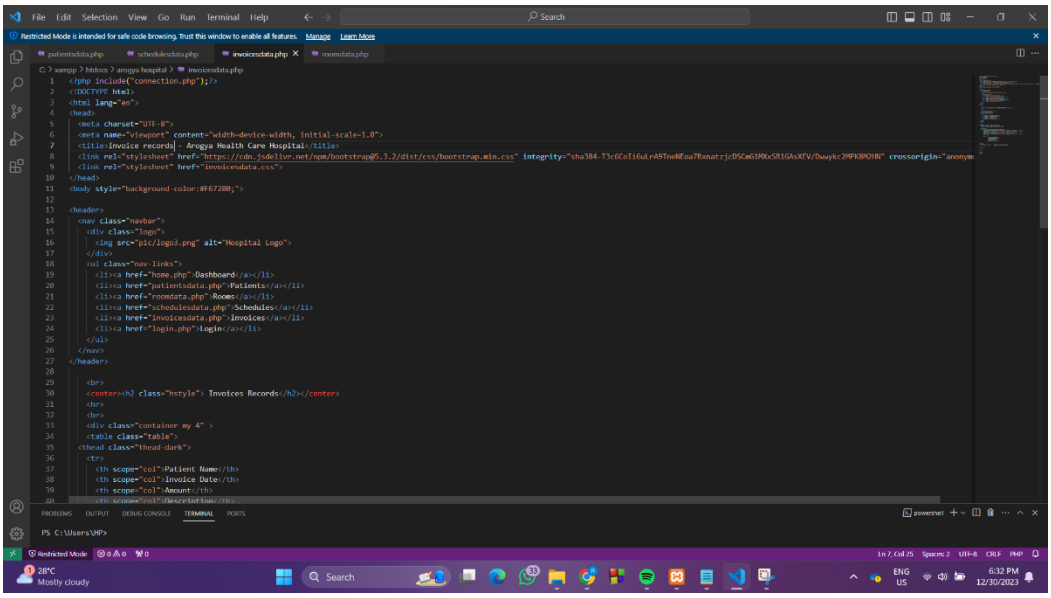


Figure 62 : Invoice html page

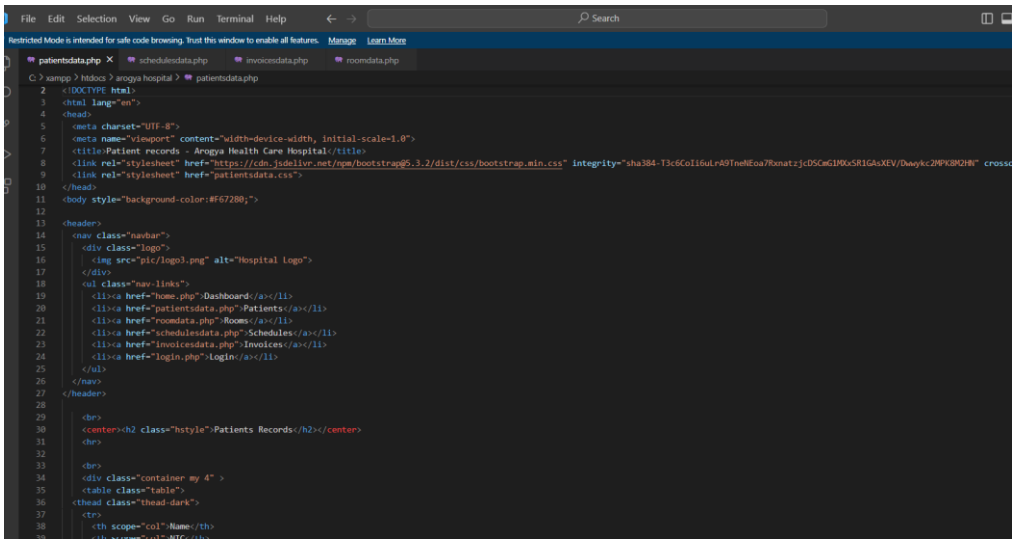


Figure 63 : patient html page

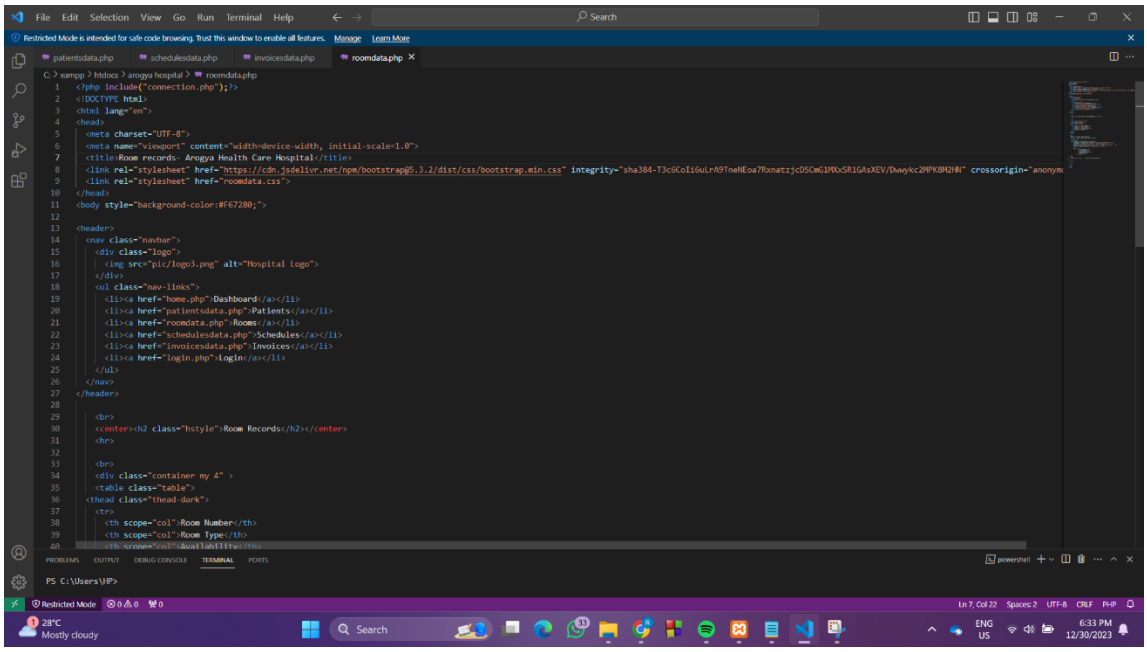


Figure 64 : room html page

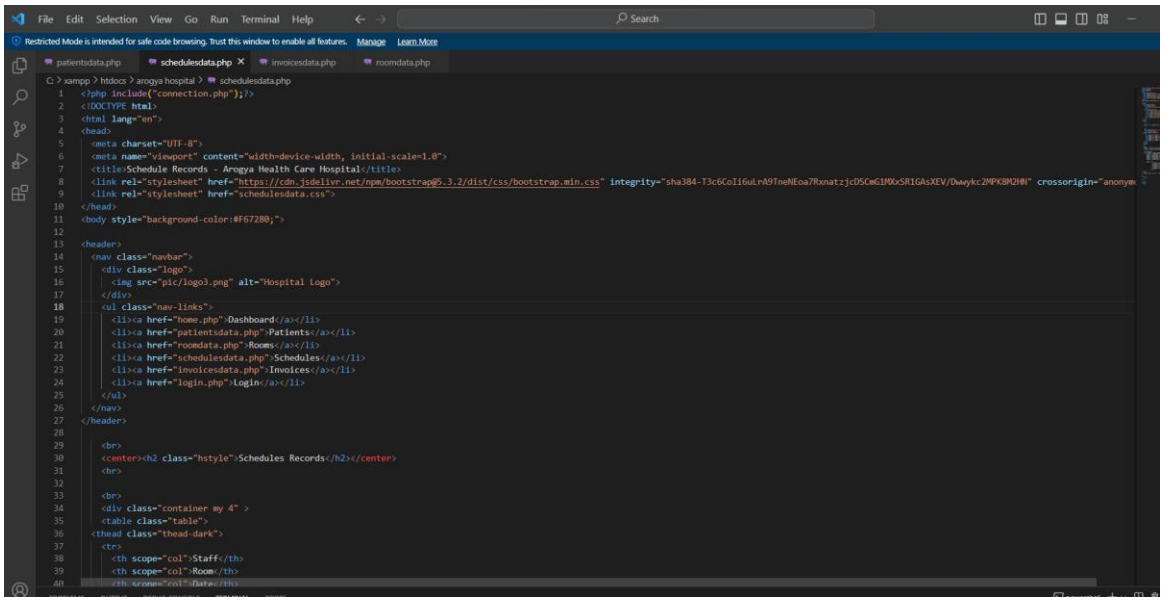


Figure 65 : Schedule html page

- CSS

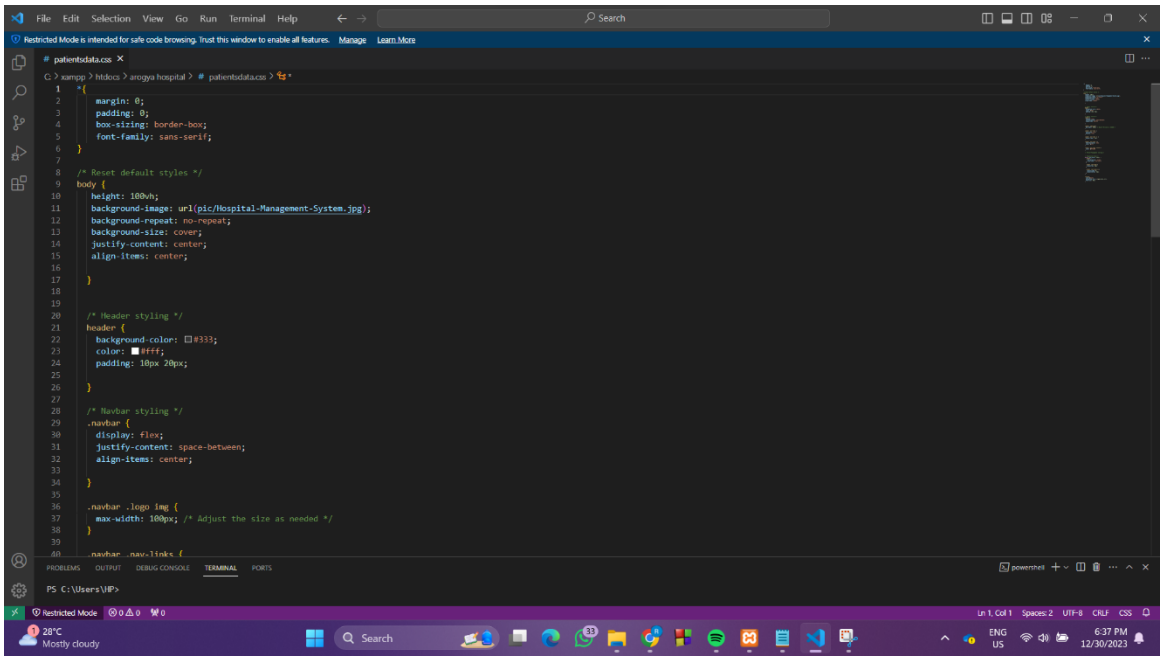


Figure 66 : Patients css 1

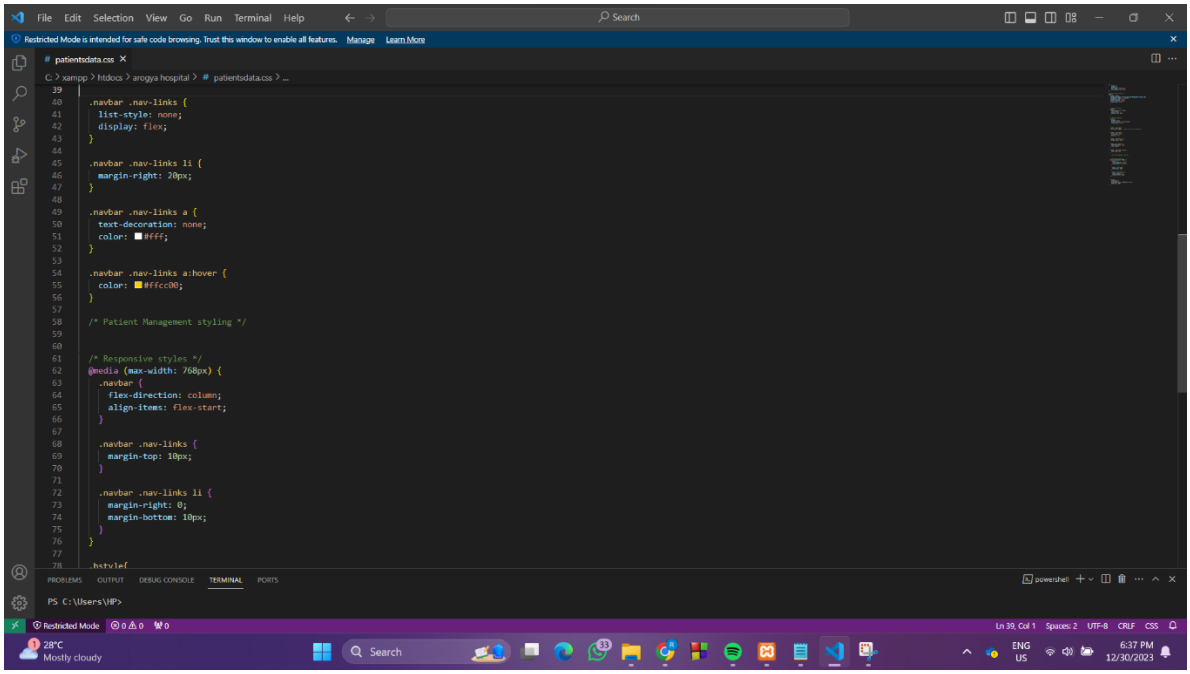


Figure 67 : patient css 2

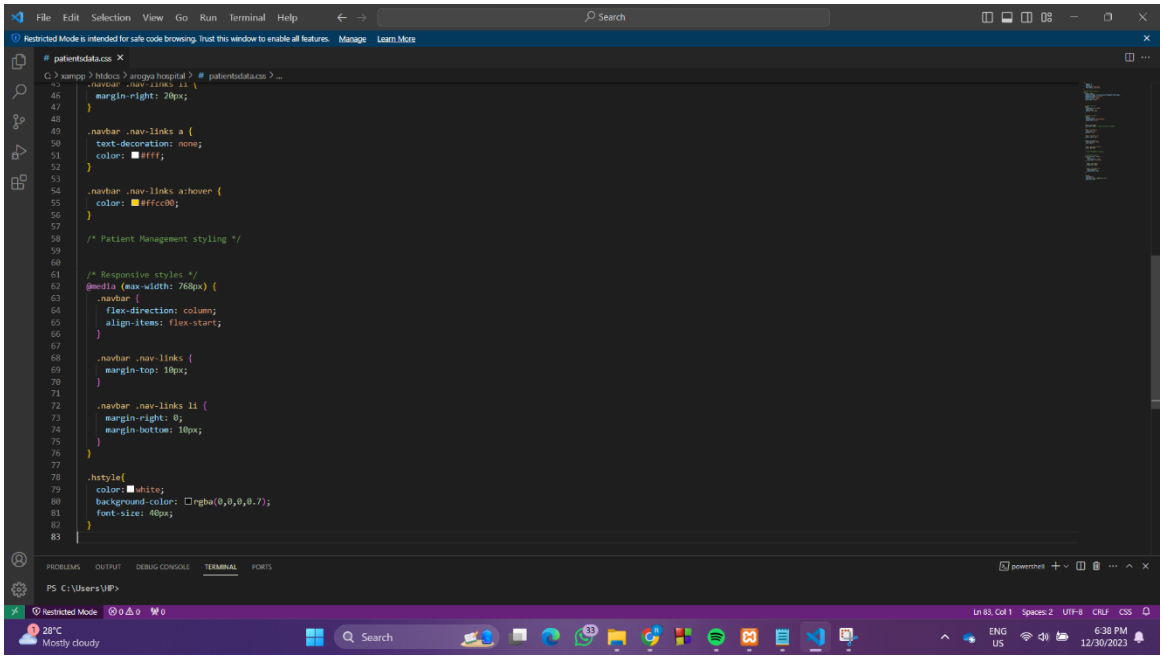


Figure 68 : patient css 3

- PHP

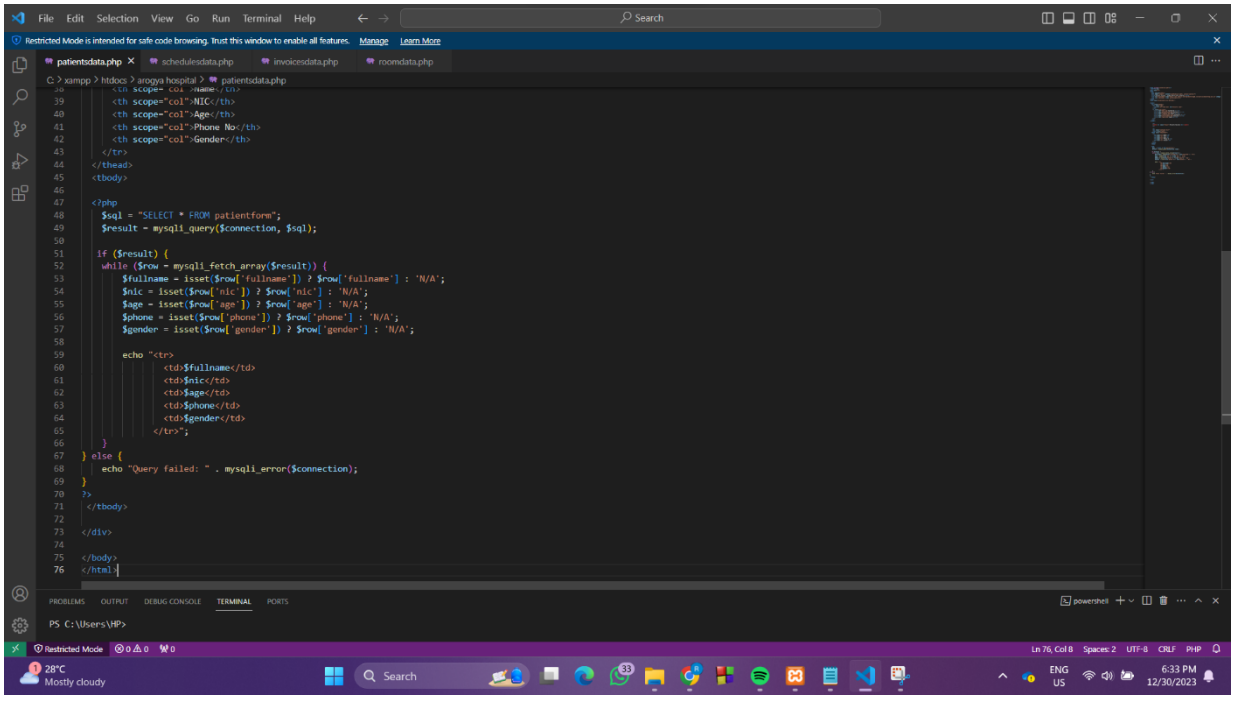


Figure 69 : Patients php page

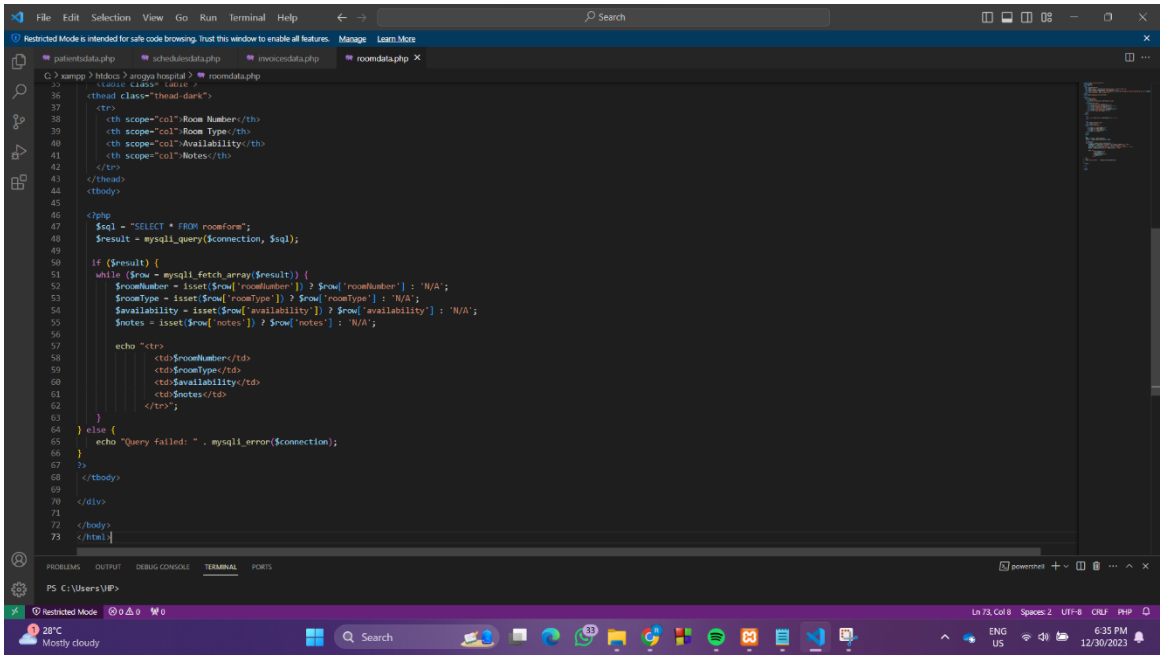


Figure 70 : Room php

Interface Codes

- HTML

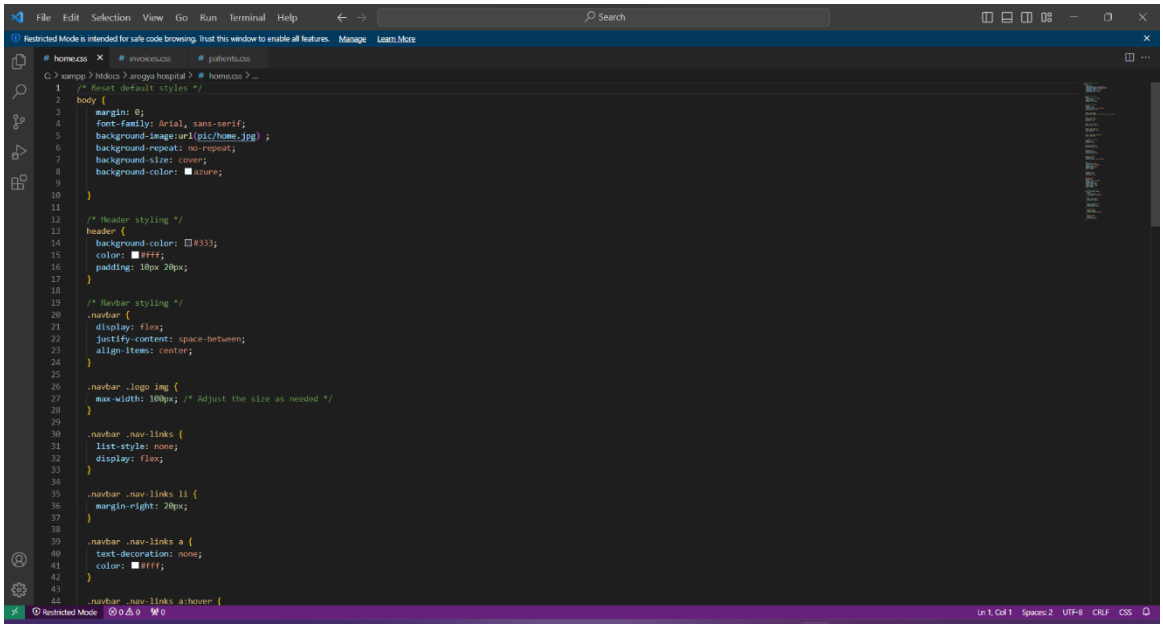
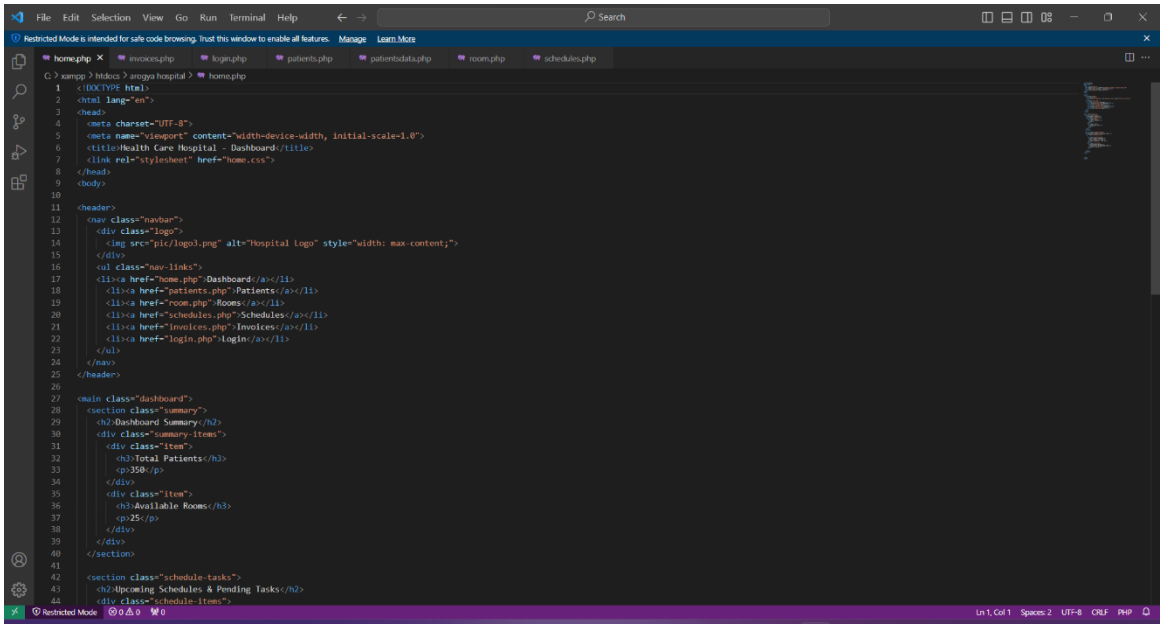
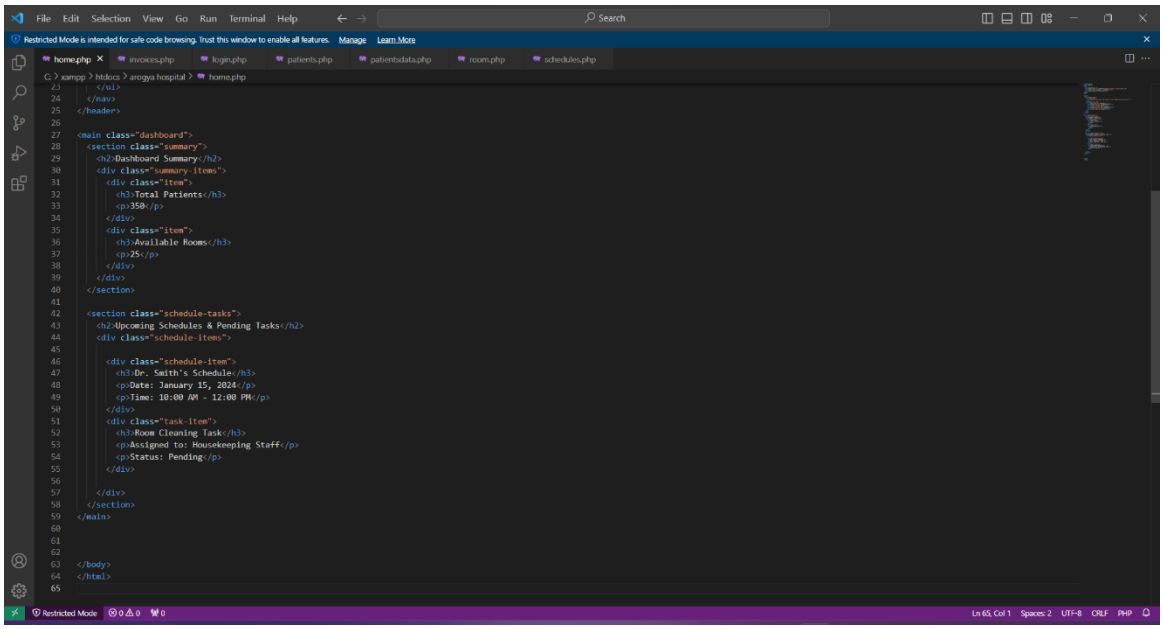


Figure 71 : Home html 1



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Health Care Hospital - Dashboard</title>
  <link rel="stylesheet" href="home.css">
</head>
<body>

<header>
  <nav class="navbar">
    <div class="logo">
      <img src="pic/logo3.png" alt="Hospital Logo" style="width: max-content;">
    </div>
    <ul class="nav-links">
    <li><a href="home.php">Dashboard</a></li>
      <li><a href="patients.php">Patients</a></li>
      <li><a href="room.php">Rooms</a></li>
      <li><a href="schedules.php">Schedules</a></li>
      <li><a href="invoices.php">Invoices</a></li>
      <li><a href="login.php">Login</a></li>
    </ul>
  </nav>
</header>

<main class="dashboard">
  <section class="summary">
    <h2>Dashboard Summary</h2>
    <div class="summary-items">
      <div class="item">
        <h3>Total Patients</h3>
        <p>350</p>
      </div>
      <div class="item">
        <h3>Available Rooms</h3>
        <p>25</p>
      </div>
    </div>
  </section>

  <section class="schedule-tasks">
    <h2>Upcoming Schedules & Pending Tasks</h2>
    <div class="schedule-items">
```

Figure 72 : Home html 2



```
    </ul>
  </nav>
</header>

<main class="dashboard">
  <section class="summary">
    <h2>Dashboard Summary</h2>
    <div class="summary-items">
      <div class="item">
        <h3>Total Patients</h3>
        <p>350</p>
      </div>
      <div class="item">
        <h3>Available Rooms</h3>
        <p>25</p>
      </div>
    </div>
  </section>

  <section class="schedule-tasks">
    <h2>Upcoming Schedules & Pending Tasks</h2>
    <div class="schedule-items">

      <div class="schedule-item">
        <h3>Dr. Smith's Schedule</h3>
        <p>Date: January 15, 2024</p>
        <p>Time: 10:00 AM - 12:00 PM</p>
      </div>
      <div class="task-item">
        <h3>Room Cleaning Task</h3>
        <p>Assigned to: Housekeeping Staff</p>
        <p>Status: Pending</p>
      </div>

    </div>
  </section>
</main>


</body>
</html>

```

Figure 73 :Home html 3

- CSS

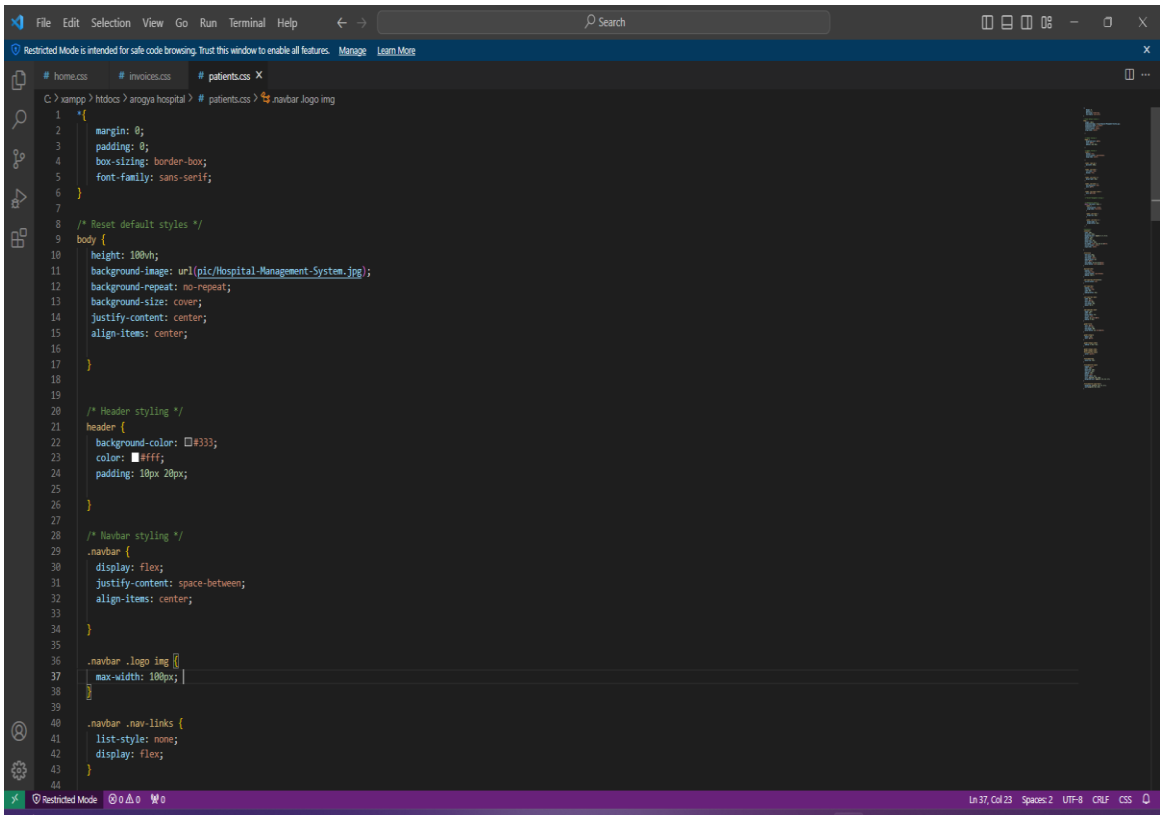


Figure 74 : patient css 1

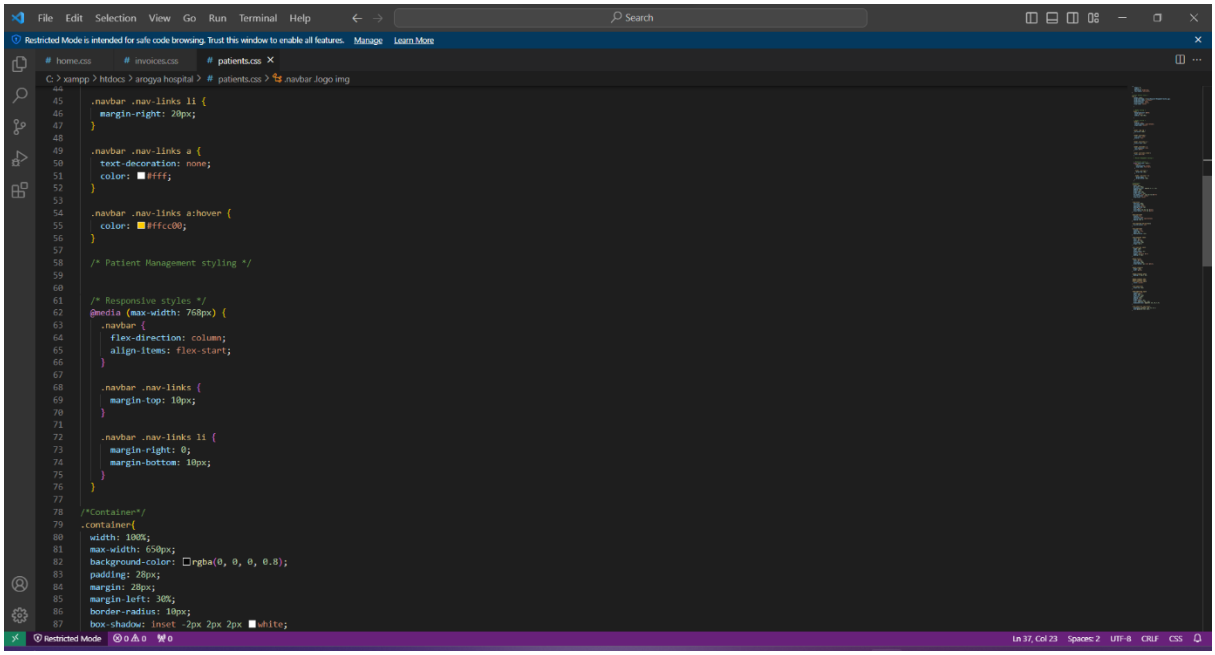
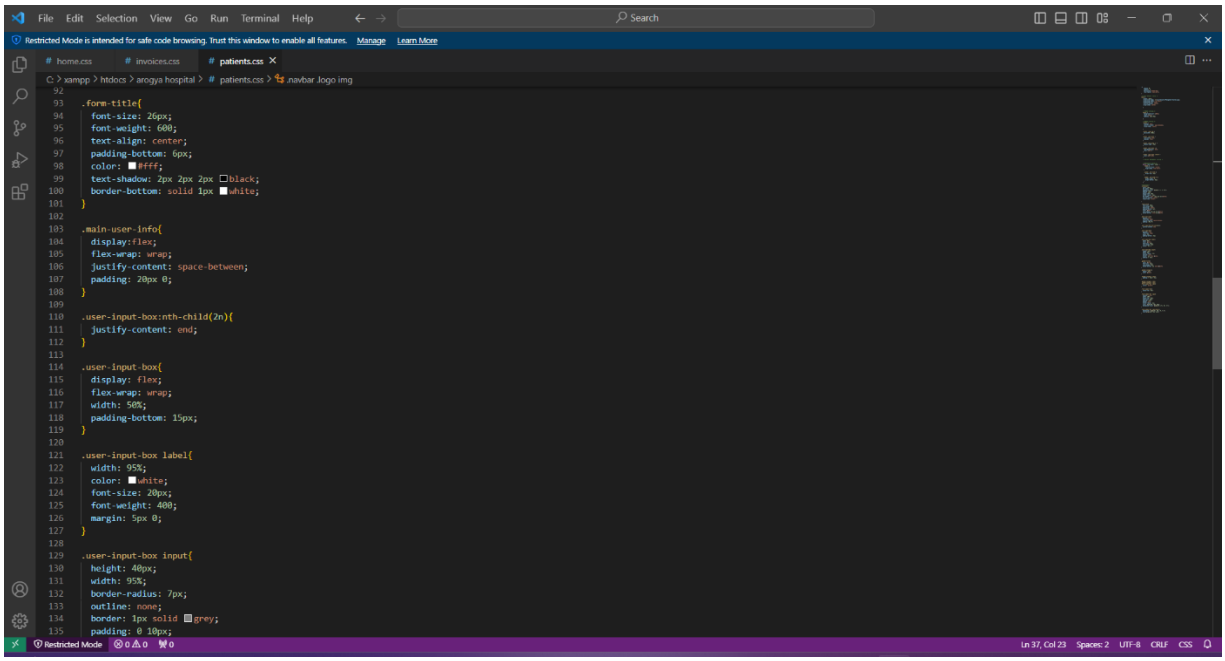
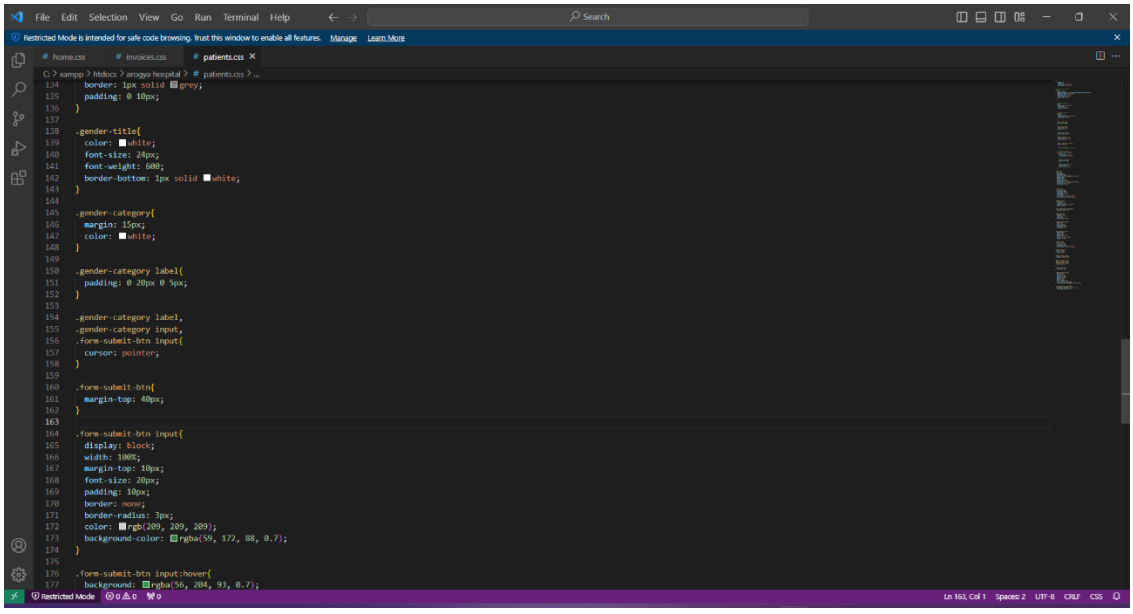


Figure 75 : Patients css 2



```
.form-title{
  font-size: 26px;
  font-weight: 600;
  text-align: center;
  padding-bottom: 6px;
  color: #fff;
  text-shadow: 2px 2px 2px black;
  border-bottom: solid 1px white;
}

.main-user-info{
  display:flex;
  flex-wrap: wrap;
  justify-content: space-between;
  padding: 20px 0;
}

.user-input-box:nth-child(2n){
  justify-content: end;
}

.user-input-box{
  display: flex;
  flex-wrap: wrap;
  width: 50%;
  padding-bottom: 15px;
}

.user-input-box label{
  width: 95%;
  color: white;
  font-size: 20px;
  font-weight: 400;
  margin: 5px 0;
}

.user-input-box input{
  height: 40px;
  width: 95%;
  border-radius: 7px;
  outline: none;
  border: 1px solid grey;
  padding: 0 10px;
```

Figure 76: patient css 3



```
  border: 1px solid grey;
  padding: 0 10px;
}

.gender-title{
  color: white;
  font-size: 24px;
  font-weight: 600;
  border-bottom: 1px solid white;
}

.gender-category{
  margin: 15px;
  color: white;
}

.gender-category label{
  padding: 0 20px 0 5px;
}

.gender-category label,
.gender-category input,
.form-submit-btn input{
  cursor: pointer;
}

.form-submit-btn{
  margin-top: 40px;
}

.form-submit-btn input{
  display: block;
  width: 100%;
  margin-top: 10px;
  font-size: 20px;
  padding: 10px;
  border: none;
  border-radius: 3px;
  color: rgb(209, 209, 209);
  background-color: rgba(59, 172, 88, 0.7);
}

.form-submit-btn input:hover{
  background: rgba(56, 204, 93, 0.7);
```

Figure 77 : patient css 4

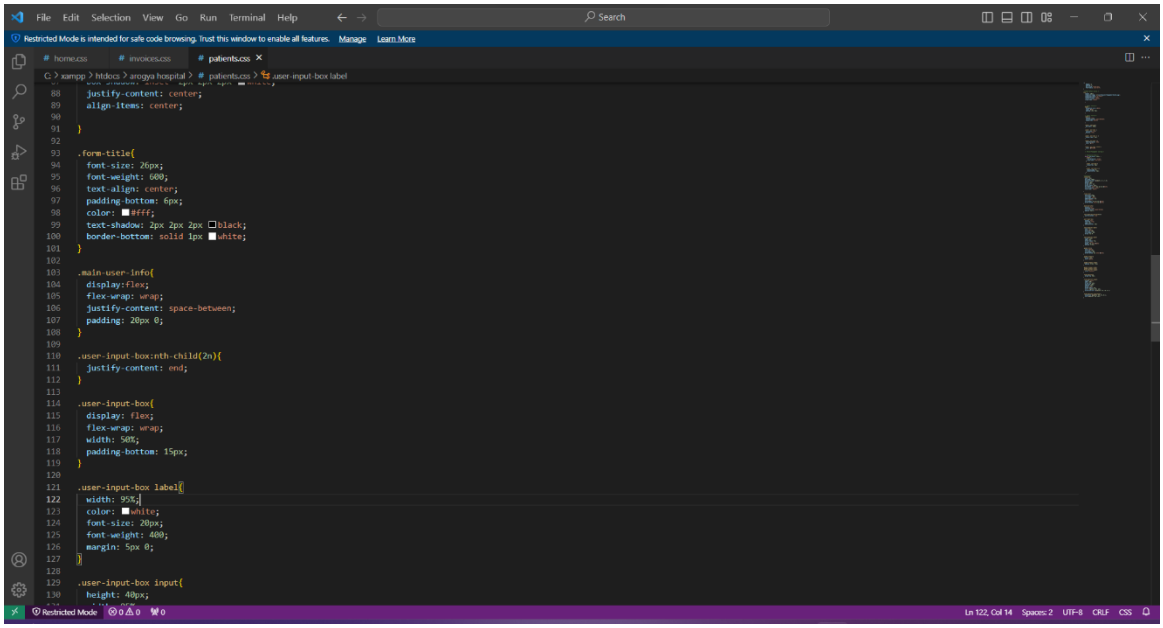


Figure 78 : patient css 5

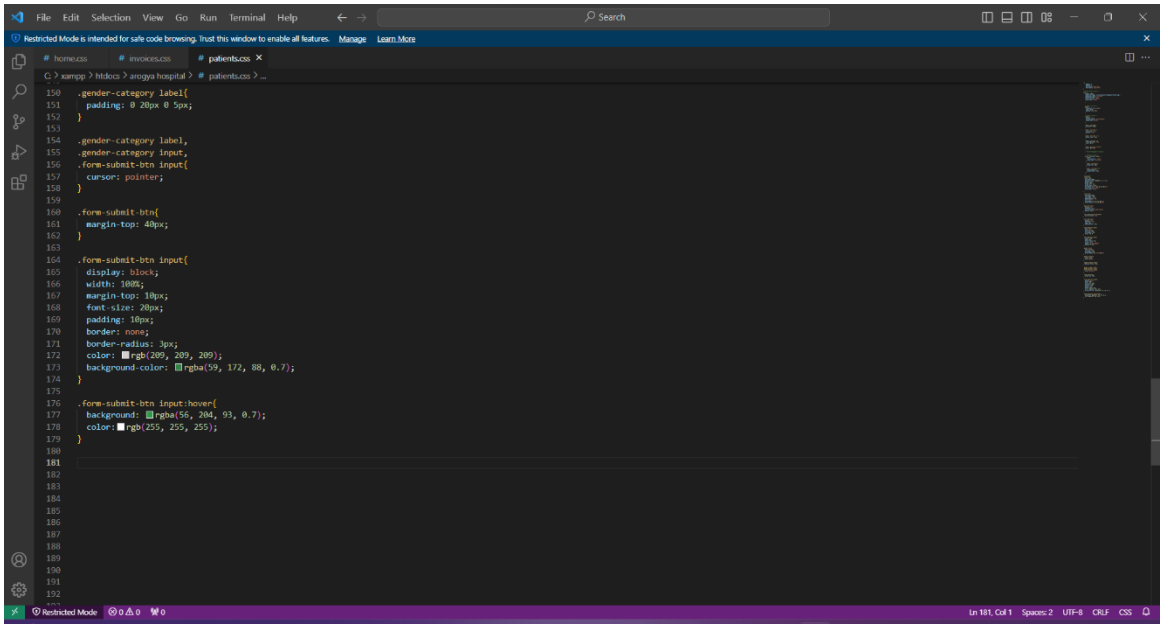


Figure 79 : patient css 6

- PHP

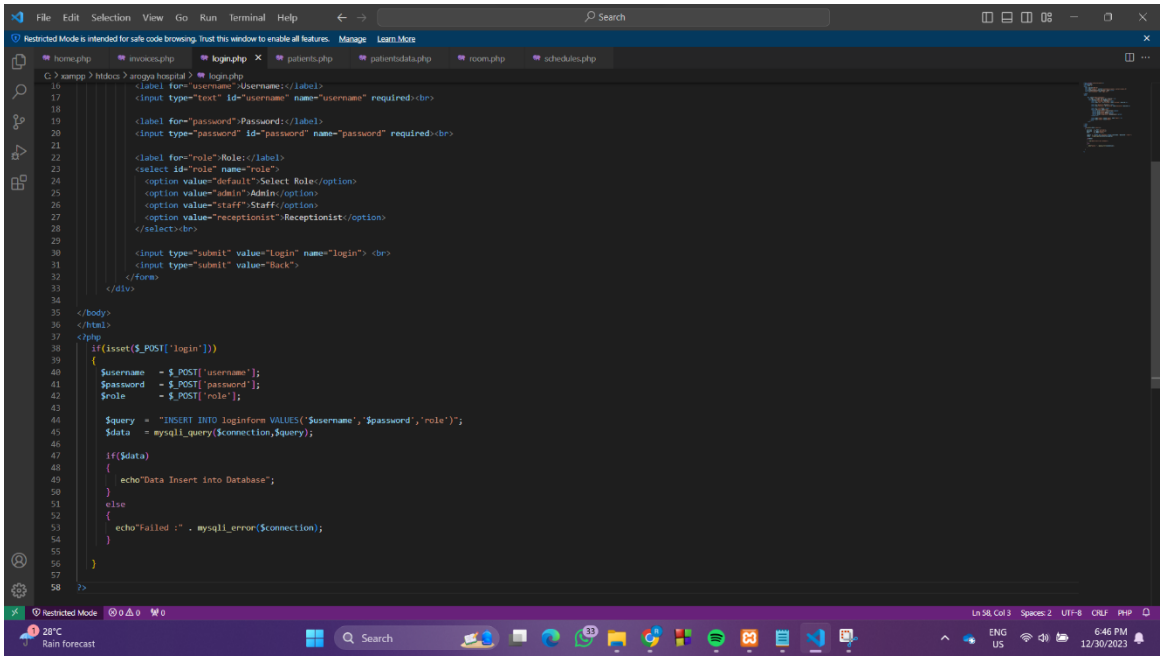


Figure 80 : login php

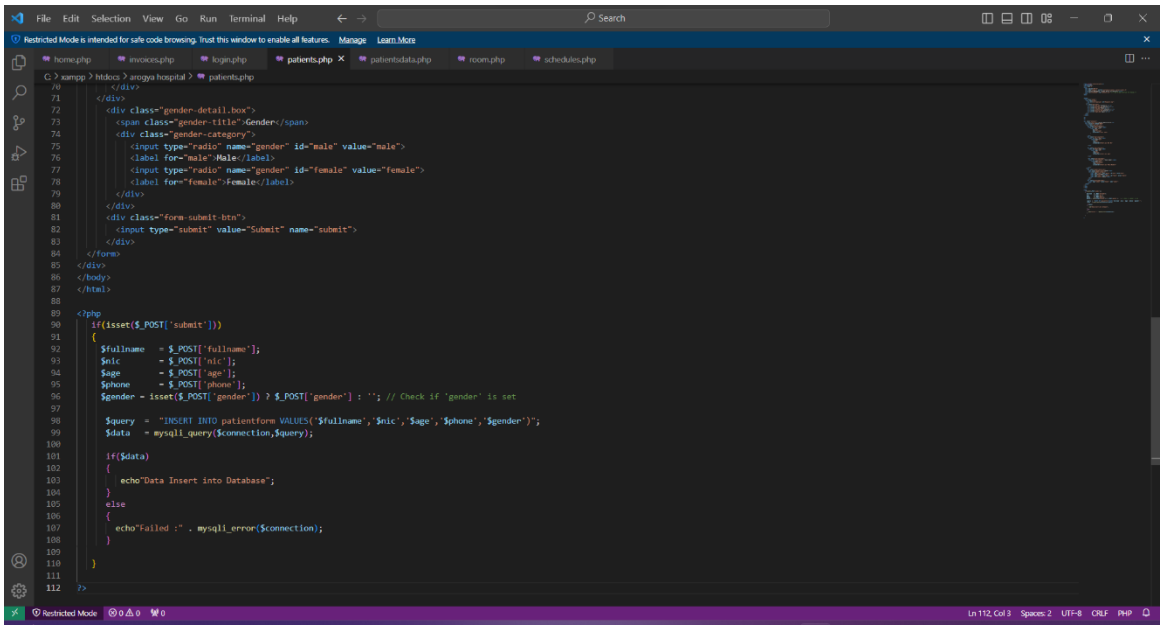


Figure 81: patients php

- Javascript

```
    <script>
      function validateForm() {
        var username = document.getElementById('username').value;
        var password = document.getElementById('password').value;

        if (username.trim() === '' || password.trim() === '') {
          alert('Please enter both username and password.');
          return false;
        }
        return true;
      }
    </script>
  </head>
  <body>
```

Figure 82 : Login java script

Databases Design for the created website for Arogya Health care hospital

This databases were created in Mysql by Author all the databases are in below.

Invoice Table

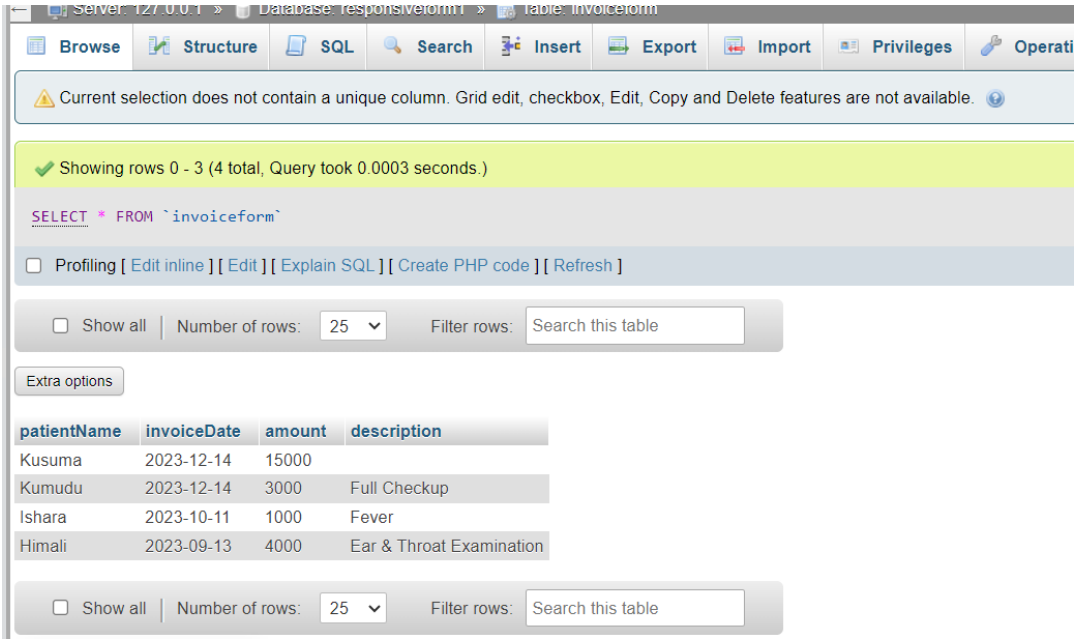


Figure 83 :Invoice table database

Login Table

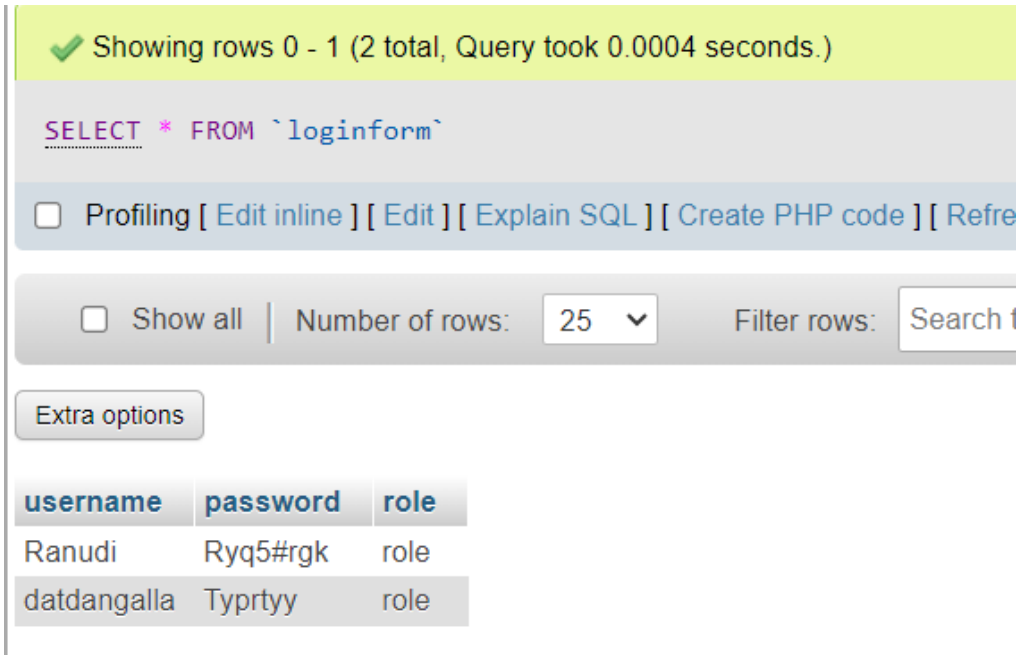


Figure 84 : Login table database

Patients Table

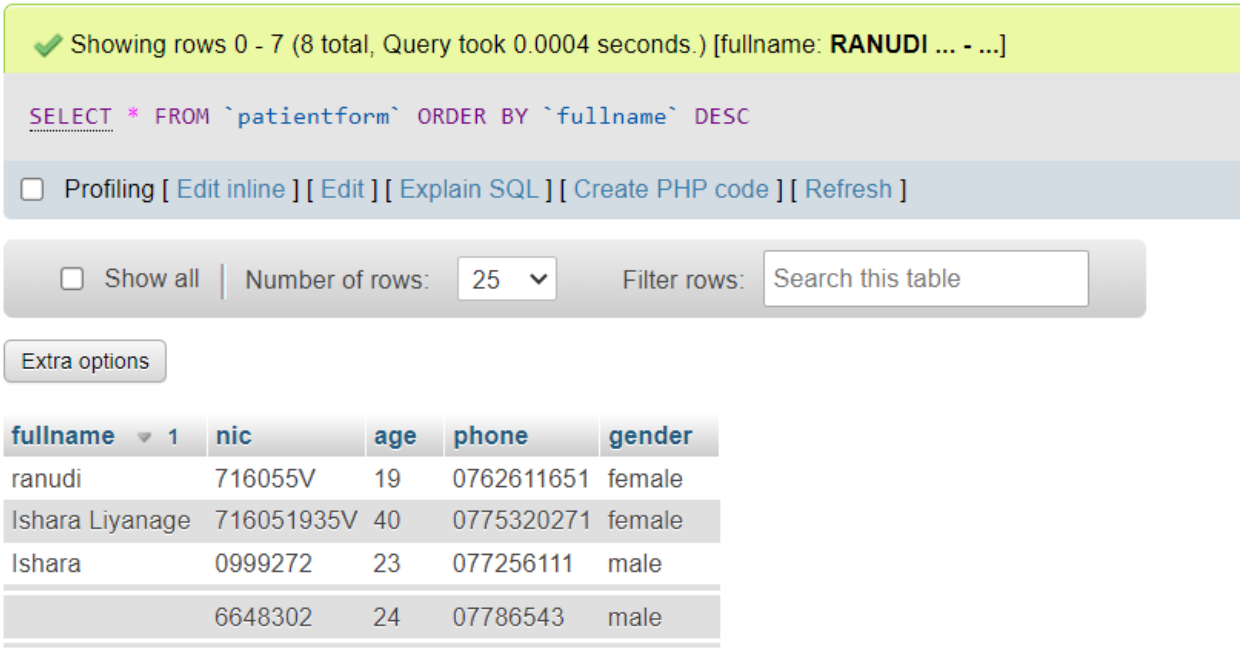


Figure 85 : Patients table database

Room table

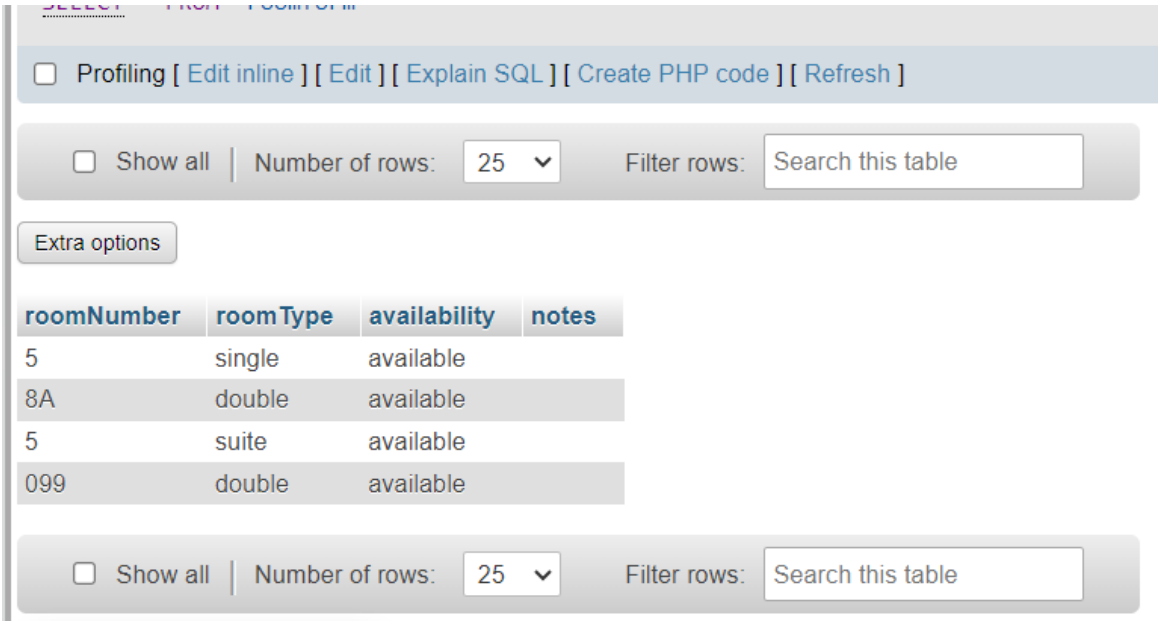


Figure 86 : room table database

Schedule table

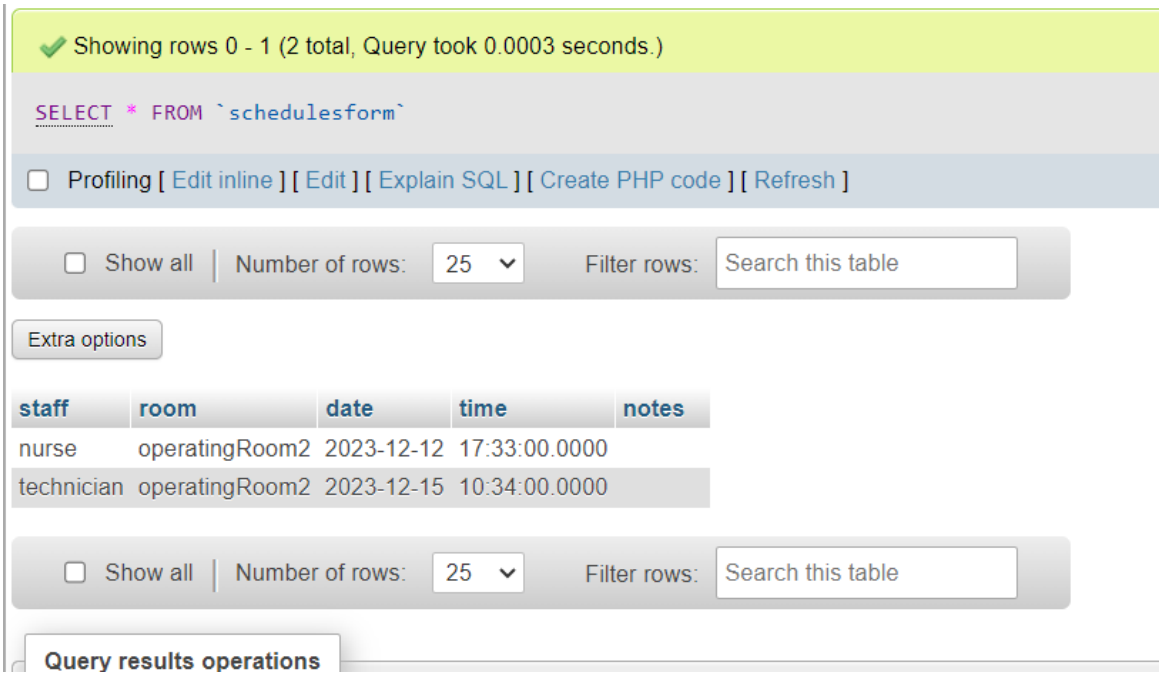


Figure 87 : schedule table database

Multiple Pages

The Arogya Healthcare website's multipage layout is essential for arranging a variety of features for effective healthcare management. The basis is patient registration, which gathers essential data for keeping thorough patient records, including patient names, NIC numbers, ages, phone numbers, and genders. By inputting room numbers and choosing between single, double, or suite categories, room availability check screens enable medical workers to quickly discover available rooms, enabling efficient room assignments. Sections of schedules help organize staff duties, making it possible to choose between staff kinds (such as doctors, nurses, and technicians) and operating rooms, and simplifying day-to-day operations. Furthermore, by enabling the entry of patient names, invoice dates, charges, and descriptions, invoice creation pages promote smooth invoicing and guarantee clear financial documentation. Informational sections provide details about hospital policies, contact information, and FAQs.

Navigation

The HMS website of Arogya Health Care Center has an easy-to-use navigation system with distinct menus and frameworks that make it simple to go between different parts. Organized menu designs provide quick access to a variety of features, making it easy for users to handle patient information, room assignments, appointments, and payment. In addition to navigation, breadcrumb trails offer users easy backward navigation and contextual signals for system orientation. The user experience is improved by this well-organized navigation, which makes it easier to navigate between the various HMS components.

Consistent Design

A consistent and well-coordinated user interface that is customized to Arogya Health Care's brand identity is ensured by the consistency of design components throughout all pages of the HMS website. Maintaining consistency in font, color schemes, layout, and user interface elements helps create a visually identifiable brand. Users will feel more familiar with the design, which makes it easier for them to recognize and navigate between the different areas. User engagement and confidence in the hospital's digital platform are improved by the uniform design approach.

Responsive Design

To accommodate a range of customer preferences, the HMS website of Arogya Health Care Center must be responsive across numerous devices. The website's material adjusts to different screen sizes without any difficulty, whether it is viewed on PCs, tablets, or mobile phones. This flexibility guarantees a consistent user experience by maintaining functionality and aesthetic appeal across devices. Adaptable design and information presentation improve usability and accessibility for a variety of users.

High-quality Content

All of the pages on the Arogya Health Care Center's HMS website have accurate, pertinent, and high-quality content. Information is presented clearly, accurately, and in a way that is directly related to its intended use in all areas of the business, from patient registration to scheduling and billing. Users are guided through each step of the process with ease with clear instructions and input areas. Adding crisp photos, flow charts, or educational films to textual material enhances understanding and participation. This dedication to producing top-notch content improves consumer pleasure and interaction.

Search Engine Optimization (SEO)

Improving visibility and reach of the HMS website requires optimizing it for search engines. Every page has carefully constructed descriptions, metadata, and targeted keywords that correspond with healthcare services. Search engine indexing is aided by structured data markup, which increases the discoverability of content. Furthermore, local SEO tactics and mobile optimization that leverage location-based data and geographic keywords improve visibility even further and draw in relevant local users looking for healthcare services.

Interactivity

Real-time updates and user input forms on the website promote interactivity. Users have platforms for active engagement through input forms embedded inside various parts, such as patient registration, room availability checks, scheduling, and invoice production. Real-time updates guarantee that users are informed promptly of any changes in room availability or scheduling, which promotes effective communication and decision-making.

Users are empowered by this interactive framework, which increases their engagement in overseeing hospital operations.

Compare and contrast the multipage website to the Arogya Health Care Center

	Login	Dashboard	Patients	Rooms	Schedules	Invoices
User Friendly web pages	Yes	Yes	Yes	Yes	Yes	Yes
Can be used by non-functionalities	Yes	Yes	Yes	Yes	Yes	Yes
Fast responding	Yes	Yes	Yes	Yes	Yes	Yes
Responsive Design	Yes	Yes	Yes	Yes	Yes	Yes
Attractive Design	Yes	Yes	Yes	Yes	Yes	Yes
Ability to log in within the given username and password	Yes	No functions given	No functions given	No functions given	No functions given	No functions given
Ability to enter information to given fields	No functions given	No functions given	Yes	Yes	Yes	Yes
Ability to insert records	No functions given	No functions given	Yes	Yes	Yes	Yes
Ability to update recordes	No functions given	No functions given	Yes	Yes	Yes	Yes
Ability to delete recordes	No functions given	No functions given	Yes	Yes	Yes	Yes

Table 16 : Compare and contrast the multipage website to the Arogya Health Care Center

The technologies and tools used in the creation of Arogya Health Care website

Using Figma for Wireframes: The HMS website's early wireframes and design mockups were created using Figma, a potent design and prototyping tool. The Apex Design Works team was able to brainstorm and conceptualize the website's design, user interface components, and page capabilities thanks to its facilitation of collaborative design work. The user-friendly interface of Figma made it possible to create dynamic wireframes, which acted as a guide for the design and organization of the website.

MySQL for Databases: A dependable and popular open-source relational database management system, MySQL was utilized for data administration and storage. Critical hospital data, such as patient records, staff profiles, room schedules, and invoice data, were handled by MySQL with efficiency. Its scalability, dependability, and strong data management skills make it the perfect option for storing and retrieving many types of data inside the HMS.

The foundational languages for web development, HTML5 and CSS3, were used in Visual Studio to construct the layout and design of the HMS website. For HTML and CSS coding, Visual Studio offered an integrated development environment (IDE) that was both feature-rich and adaptable. The website's fundamental components were established by HTML5, while style and visual enhancements were added by CSS3. This combination ensured an aesthetically pleasing and intuitive interface for handling hospital information.

PHP in XAMPP: XAMPP is a well-liked cross-platform web server solution stack that was used to integrate PHP, a server-side scripting language, into the HMS. The website's dynamic features were made possible with PHP, which also allowed for form management, database interactions, and server-side processing. With PHP integrated with Apache, MySQL, and other components, XAMPP offered a local development environment that made it easier to test and implement the PHP-based features of the HMS.

While creating the HMS for Arogya Health Care, Figma improved the first design phase, MySQL handled vital hospital information, Visual Studio helped with creating HTML and CSS for the website's layout and appearance, and XAMPP combined with PHP enabled

dynamic server-side features. A strong, intuitive, and useful hospital management system that is customized to meet the unique requirements of Arogya Health Care was created mainly to the combined efforts of these technologies and tools.

The technical challenges faced By Athor and The way Author overcome the them

Arogya Health Care faced a number of technological obstacles when building the Hospital Management System (HMS), all of which needed to be strategically resolved to guarantee the system's effective deployment.

1. Data Centralization and Management

Challenge :The challenge of data centralization and management. Previously, Arogya Health Care used manual methods with dispersed data on paper forms and storage systems. Because of the possibility of inconsistencies and insufficient information, integrating and centralizing this dispersed data presented a considerable difficulty.

Solution: A careful data migration plan was developed in order to get over this obstacle. To combine data into a single MySQL database, meticulous extraction, transformation, and loading (ETL) procedures were needed. To guarantee data integrity and compliance with management requirements, validation scripts were created. Prior to migration, inconsistencies were also resolved with the use of manual verification and cross-referencing.

2. User Interface Design and Functionality

Challenge: User Interface Design and Functionality It was difficult to create a user interface (UI) that was both useful and intuitive, accommodating a variety of functions like staff management, room scheduling, and patient registration, all while guaranteeing ease of navigation. Achieving a balance between usefulness and an intuitive interface was essential.

Solution: Iterative design modifications were made possible by using Figma for wireframing. In order to provide a balance between functionality and user accessibility, the UI was refined with the use of user testing and feedback loops. To simplify user interaction between different HMS sections, the UI was created with straightforward forms, obvious navigation paths, and visual hints.

3. Database Integration and Performance

Challenge: Because of the volume of data handled in real-time, it was difficult to integrate MySQL databases with the PHP-driven backend in a way that ensured optimal performance and data consistency.

Solution: To improve performance, database optimization techniques such as normalization, indexing, and query optimization were used. Additionally, latency was reduced and system responsiveness was enhanced by using PHP scripts for effective data retrieval, manipulation, and updating of records. Monitoring and maintenance of databases on a regular basis helped to reduce performance bottlenecks.

4. Security and Privacy Compliance

Challenge: Because medical data is vital, it can be difficult to maintain strong security measures to protect private patient information and to comply with laws governing patient privacy, such HIPAA.

Solution: Strict security protocols, such as role-based access controls, secure authentication methods, and encryption for data in transit and at rest, were put into place to strengthen data security. Compliance with healthcare privacy standards was guaranteed by regular security audits, vulnerability assessments, and staff training on data handling methods.

5. Testing and Quality Control

Challenge: Determining and fixing faults, usability problems, and performance problems required a thorough evaluation of the HMS across many functionality, devices, and scenarios.

Solution: Unit, equality, user acceptability, and stress testing were all included in the stringent testing processes that were set forth. Automatic testing tools and test-driven development approaches made problem discovery and fixing easier. The system was iteratively improved through ongoing feedback loops from stakeholders.

Overcoming challenges to provide an efficient and secure hospital management solution, the HMS for Arogya Health Care was successfully implemented by tackling these technical issues by careful design, continuous development, thorough testing, and ongoing improvement.

Test Plan to review the performance and design of a multiple pages

Activity	Test Case	Test Data	Expected Result	Actual Result	Status
1	Checking the login	Relevant user name password and role	Login suceessful	Login successful	pass
2	Check the patient registration	Information of a random person	Inserted and show the data in records	Inserted suceessfully and show the data in records	pass
3	Check the delete in patient records	Information of a random person	Deleted	Deleted sucessfully	pass

4	Submit availability the room	Information of a random Person	Available	Available	Pass
5	Generate a invoice	Information from random person	Generate an invoice	Successfully generated and show the data in recordes	pass
6	Checking login	Incorrect username	login faild	Login faild	pass
7	Checking login	Incorect password	login faild	Login faild	pass
8	Checking login	Incorect Role	login faild	Login faild	pass

Table 17 : Test Plan to review the performance and design of a multiple pages

Quality Assurance (QA) process

In web application development, quality assurance (QA) is a methodical procedure meant to guarantee that software products fulfill requirements, satisfy quality standards, and provide the best possible user experiences. Throughout the development lifecycle, this process consists of multiple stages and approaches to detect problems, improve functionality, and uphold high standards.

Requirement analysis, which is the foundation of the quality assurance process, establishes a comprehensive grasp of project specifications, user demands, and functional expectations. This first stage lays the foundation for creating an extensive test plan that describes the goals, approaches, resources, and scope of the testing activities.

Detailed test cases, scenarios, and scripts are created based on the specified requirements during test planning. They act as a guide for carrying out several kinds of testing, such as performance, usability, security, and functional evaluations. Thorough testing guarantees that the program satisfies the required requirements and performs as expected under various circumstances.

Any errors or inconsistencies found during the testing process are carefully recorded and reported. Problems are resolved more quickly when QA and development teams work together. To ensure successful bug fixes and preserve system stability, defects are fixed and extensive retesting is carried out.

Regression testing is essential to the quality assurance process since it confirms that recent modifications or fixes have no adverse effects on the application's previously functional components. In order to make sure the application is in line with user demands, user acceptability testing, or UAT, entails involving stakeholders or end users in the validation process against their expectations and requirements.

The most important aspect of quality assurance is thorough reporting and documentation, which offers in-depth information on test strategies, cases, outcomes, and defect reports.

These documents support decision-making processes and act as references for upcoming enhancements.

One of the main tenets of QA is continuous improvement. Throughout the process, feedback is received and used to improve testing efficacy overall, update test cases, and optimize techniques. The application is always evolving to suit changing requirements and uphold high standards thanks to the iterative nature of quality assurance.

The way of QA process was applied to the Arogya Health Care Hospital

Several methodical procedures were followed when using the Quality Assurance (QA) process to guarantee that the Hospital Management System (HMS) for Arogya Health Care matched predetermined quality standards and the unique requirements of the healthcare context.

Requirement Analysis: The QA process started with a thorough examination of the unique requirements of Arogya Health Care as well as the functional requirements needed for the HMS. It was crucial to comprehend the operational nuances, staff scheduling, patient management, processes, and operational details of the hospital in order to precisely customize the system to meet their needs.

Test Design and Planning: A thorough test strategy was created using the criteria that were gathered. The strategy specified the extent, goals, approaches, and assets required for comprehensive examination. Carefully considered test cases and scenarios were created to cover a range of HMS topics, including as staff scheduling, patient registration, room assignment, and invoice production.

Test Environment Setup: Specialized testing environments that replicated the operational configuration of the hospital were created. These environments supplied a controlled testing environment and were essential for accurate evaluations without affecting the live systems.

Test Case Execution: The QA team tested the HMS's functionality, usability, security, and performance in a rigorous manner, following the guidelines provided in the test cases. In order to guarantee that the systems for managing patient information, assigning rooms, scheduling staff, and billing operated without a hitch, functionality was verified.

Defect Identification and Resolution: All inconsistencies or problems found during testing were meticulously recorded and reported. In order to quickly address issues that were raised, the QA team worked closely with the development teams. In order to guarantee successful bug repairs, defects were fixed and extensive retesting was carried out.

User Acceptance Testing (UAT) and Regression Testing: Regression testing was done to make sure that new updates or fixes did not adversely affect the HMS's current functionalities. In order to verify that the system satisfies their needs and ensures that usability and functionality are in line, user acceptability testing entailed including hospital stakeholders and end users.

Documentation and Continuous Improvement: A thorough documentation was created, comprising defect reports, test cases, test plans, and test results. These records acted as a resource for upcoming enhancements and well-informed decision-making procedures. Feedback received during testing was put to use for improving test cases, testing procedures, and general efficacy in later iterations.

Test Case Evaluation

- Login Functionality

The login mechanism exhibited consistent success, ensuring secure access control. It effectively denied unauthorized access attempts with incorrect credentials, demonstrating robust security measures.

- Patient Registration

The functionality to insert and display patient data functioned effectively, reflecting the system's capability to manage patient information efficiently. Successful registration and retrieval validate the system's data handling.

- Delete in Patient Records

The deletion of patient records was executed seamlessly, affirming the system's data management capabilities. The successful removal of records ensures efficient maintenance of patient data.

- Room Availability Submission

Submitting room availability for random individuals showed successful implementation, indicating proper room management functionalities. The system accurately managed and updated room availability information.

- Invoice Generation

While the test case suggests successful invoice generation, verification of the actual generation is necessary. Re-evaluation and validation are crucial to ensure the accuracy and completeness of this critical functionality.

- Login Failure Scenarios

The system appropriately restricted access during login failure scenarios, confirming its ability to thwart unauthorized entry. Failed login attempts with incorrect inputs validated the system's robust security measures.

Successes

- Functionalities

The core functionalities of patient registration, room management, and login mechanisms exhibited successful implementation. They showcased efficiency in handling critical hospital operations.

- Security Measures

The system's ability to prevent unauthorized access during login failures highlighted its strong security protocols. This robust control mechanism ensures data integrity and access control.

Recommendations for Improvement

- Invoice Generation Test Case

Re-evaluate the invoice generation process to validate accuracy. Ensure verification of invoice generation to guarantee precise billing and financial record management.

- Comprehensive Testing

Expand the testing scope to encompass various scenarios, including edge cases and stress testing. This will reveal potential vulnerabilities and areas needing enhancement.

- Logging and Reporting

Enhance logging mechanisms to capture detailed system interactions and errors encountered during testing. Improved logging aids in better debugging and troubleshooting.

- User Feedback Integration

Incorporate mechanisms to gather user feedback for insights into user experience and system usability, facilitating further enhancements aligned with user needs.

- Documentation

Ensure comprehensive documentation of test cases, results, and system improvements. This documentation serves as a reference for future enhancements and continuous improvement.

Overall Assessment

The HMS development for Arogya Health Care showcased successful functionalities in critical areas like patient management and access control. However, thorough testing, particularly in invoice generation, and improvements in logging mechanisms and user feedback integration are necessary. Addressing these recommendations will augment the system's reliability, functionality, and overall user satisfaction, aligning it more closely with Arogya Health Care's operational needs.

References

BrowserStack. (n.d.). How to build a website using HTML and CSS. [online] Available at: https://www.browserstack.com/guide/build-a-website-using-html-css. [Accessed 12 Dec. 2023].

Churt, R. (2017). How to Audit Your Website for Improved SEO and Conversions. [online] Hubspot.com. Available at: https://blog.hubspot.com/marketing/website-audit. [Accessed 19 Dec. 2023].

Kohan, B., 2022. Guide to Web Application Development. [Online] Available at: https://www.comentum.com/guide-to-web-application-development.html Stevewhims, 2021.

DNS Servers. [Online] Available at: https://docs.microsoft.com/en-us/windows/win32/dns/dns-servers

Hamilton, T. (2019). Web Application Testing: 8 Step Guide to Website Testing. [online] Guru99.com. Available at: https://www.guru99.com/web-application-testing.html. [Accessed 24 Dec. 2023].

Grading Rubric

Grading Criteria	Achieved	Feedback
LO1 Explain server technologies and management services associated with hosting and managing websites		
P1 Identify the purpose and types of DNS, including explanations on how domain names are organized and managed.		
P2 Explain the purpose and relationships between communication protocols, server hardware, operating systems and web server software with regards to designing, publishing and accessing a website.		
M1 Evaluate the impact of common web development technologies and frameworks with regards to website design, functionality and management.		
M2 Review the influence of search engines on website performance and provide evidence-based support for improving a site's index value and rank through search engine optimization.		
D1 Justify the tools and techniques chosen to realize a custom built website.		
LO2 Categories website technologies, tools and software used to develop websites		
P3 Discuss the capabilities and relationships between front-end and back-end website technologies and explain how these relate to presentation and application layers.		
P4 Discuss the differences between online website creation tools and custom built sites with regards to design flexibility, performance, functionality, User Experience (UX) and User Interface (UI).		

M3 Evaluate a range of tools and techniques available to design and develop a custom built website.		
LO3 Utilize website technologies, tools and techniques with good design principles to create a multipage website		
P5 Create a design document for a branded, multipage website supported with medium fidelity wireframes and a full set of client and user requirements.		
P6 Use your design document with appropriate principles, standards and guidelines to produce a branded, multipage website supported with realistic content.		
M4 Compare and contrast the multipage website created to the design document.		
D2 Critically evaluate the design and development process against your design document and analyse any technical challenges.		
LO4 Create and use a Test Plan to review the performance and design of a multipage website		
P7 Create a suitable Test Plan identifying key performance areas and use it to review the functionality and performance of your website.		
M5 Evaluate the Quality Assurance (QA) process and review how it was implemented during your design and development stages.		
D3 Critically evaluate the results of your Test Plan and include a review of the overall success of your multipage website; use this evaluation to explain any areas of success and provide justified recommendations for areas that require improvement.		