

| INTERNAL VERIFICATION – ASSESSMENT DECISIONS | | | | |
|--|---|-------------------|-------------|--|
| Programme title | BTEC Higher National Diploma in Computing | | | |
| Assessor | | Internal Verifier | | |
| Unit(s) | Unit 10: Web Design and Development | | | |
| Assignment title | Online Hospital Management System | | | |
| Student's name | | | | |
| 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: <ul style="list-style-type: none"> • 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 | | | |
| Unit Title | Unit 10: Website Design & Development | | |
| Assignment Number | 1 | Assessor | |
| Submission Date | | 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 P1 P2 M1 M2 D1
Descripts

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

Pass, Merit & Distinction P3 P4 M3 D1
Descripts

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

Pass, Merit & Distinction P5 P6 M4 D2
Descripts

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

Pass, Merit & Distinction P7 M5 D3
Descripts

| | | |
|-------------------------------|---------------------|-------|
| 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 page.** 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.

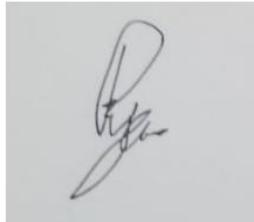
Important Points:

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 plagiarise 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.



ryandilthusha@gmail.com

1/07/2022

Student's Signature:

(Provide E-mail ID)

Date:

(Provide Submission Date)

Assignment Brief

| | |
|------------------------------|-------------------------------------|
| Student Name /ID Number | |
| Unit Number and Title | Unit 10- Web Design and Development |
| Academic Year | 2021/2022 |
| Unit Tutor | Mr. Abdur Rahman |
| Assignment Title | Online Hospital Management System |
| Issue Date | |
| Submission Date | |
| IV Name & Date | |

Submission Format:

Part 1.

Report- Submit a professional report with appropriate report formatting and guidelines followed. All the research data should be referenced along with in-text citations using Harvard referencing system.

Part 2

A fully functional web solution

Unit Learning Outcomes:

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

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

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

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

Assignment Brief and Guidance:

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 Identify 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 requirements 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 given 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 analysing 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.

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. | | |

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 :.....

Acknowledgement

I would like to express my special thanks of gratitude to my Web Design and Development lecturer Mr. Abdur Rahman sir for providing invaluable guidance and giving immense amount of knowledge to work on this assignment perfectly. I specially thanks him because he helped us in doing a lot of research and I came to know about so many new things about the database designing and developing.

Secondly, I would like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

Executive Summary

This entire assignment is based on an implementation of a fully functional system installation for Arogya Hospital. The purpose of this assignment is to improve web design and developing skills.

This report includes details about communication protocols, details about web servers and types of servers relate to web development. And details about DNS, SEO and a documentation about types of web tools and technologies.

And also, this assignment gives much more details about front end development, back end development and full stack with describing 3 layer architecture. And have compared online web creation tools and customs tools.

And also have basic database software handling, creating designs, understanding data flows, understanding different database designing methods, working with IDEs, Basic system creation using Microsoft SQL and WAMP Server and other many more technologies such as bootstrap and etc.

And used different SQL query types when creating a database for a system, make a documentation for a system with describing future enhancements for the system.

Abbreviations

CSS - Cascading Style Sheets

HTML - Hypertext Markup Language

SPAs - Single-Page web Apps

HTTP - Hypertext Transfer Protocol

FTP - File Transfer Protocol

POP - Post Office Protocol

SMTP - Simple Mail Transfer Protocol

DNS - Domain Name System

SEO - Search Engine Optimization

GSC - Google Search Console

SEO - Search Engine Optimization

ICANN - The Internet Corporation for Assigned Names and Numbers

IANA - Internet Assigned Numbers Authority

DHCP - Dynamic Host Configuration Protocol

RDBMS - Relational Database Management System

IPX - Internetwork Packet Exchange

LAN - Local Area Network

DHCP - Dynamic Host Configuration Protocol

RHEL - Red Hat Enterprise Linux

RDBMS - Relational Database Management System

QA - Quality Assurance

RAM - Random Access Memory

SMB - Server Message Protocol

NFS - Network File System

DNS - Domain Name System

OS - Operating System

SMTP - Simple Mail Transfer Protocol

IMAP - Internet Message Access Protocol

POP3 - Post Office Protocol version 3

HTTP - Hypertext Transfer Protocol

DHCP - Dynamic Host Communication Protocol

PC - Personal Computer

SDN - Software Define Networks

FTP - File Transfer Protocol

W3C - World Wide Web Consortium

IETF - Internet engineering task force

List of figures

| | |
|--|----|
| FIGURE 1. 1 USING HTTP PROTOCOL | 34 |
| FIGURE 1. 2 HACKER TRIES TO STEAL DATA THROUGH INTERNET..... | 35 |
| FIGURE 1. 3 HOW SSL ACTS WHEN INTERNET BROWSING | 36 |
| FIGURE 1. 4 FLAGGING WEBSITES THAT AREN'T SECURE..... | 36 |
| FIGURE 1. 5 FILE TRANSFERRING BETWEEN FTP SERVER AND FTP CLIENT | 37 |
| FIGURE 1. 6 USING FTP CLIENT FOR FTP | 37 |
| FIGURE 1. 7 USING STANDARD INTERNET BROWSER FOR FTP | 37 |
| FIGURE 1. 8 TYPING FTP ADDRESS IN THE WEB ADDRESS BOX..... | 38 |
| FIGURE 1. 9 FTP CLIENT INTERFACE PROPERTIES | 39 |
| FIGURE 1. 10 DRAG AND DROP FILES FROM PC TO FTP SERVER | 40 |
| FIGURE 1. 11 WEB SERVER | 41 |
| FIGURE 1. 12 WEB SERVER PROCESS IN SUMMARY..... | 42 |
| FIGURE 1. 13 WEB SERVER PROCESS | 43 |
| FIGURE 1. 14 DYNAMIC VS STATIC WEB SITES | 45 |
| FIGURE 1. 15 APACHE HTTP SERVER LOGO | 46 |
| FIGURE 1. 16 MICROSOFT INTERNET INFORMATION SERVICES (IIS) LOGO | 46 |
| FIGURE 1. 17 LIGHTTPD SERVER LOGO | 47 |
| FIGURE 1. 18 JIGSAW SERVER LOGO | 47 |
| FIGURE 1. 19 THE SUN JAVA SYSTEM LOGO..... | 48 |
| FIGURE 1. 20 PARTS OF A SERVER | 49 |
| FIGURE 1. 21 SERVER CLUSTERS..... | 55 |
| FIGURE 1. 22 TOWER SERVER..... | 56 |
| FIGURE 1. 23 RACK SERVER AND HOW RACK SERVER MOUNT IN A RACK | 57 |
| FIGURE 1. 24 BLADE SERVER AND HOW BLADE SERVER MOUNT IN AN ENCLOSURE..... | 57 |
| FIGURE 1. 25 UNDERSTAND SERVER NAME AND BRANDS OF HP PROLIANT | 58 |
| FIGURE 1. 26 DIFFERENCE BETWEEN SERVER VARIANTS..... | 59 |
| FIGURE 1. 27 UNDERSTAND SERVER NAME AND BRANDS OF DELL POWER EDGE..... | 60 |
| FIGURE 1. 28 HPE PROLIANT ML350 GEN10 | 66 |
| FIGURE 1. 29 DELL POWEREDGE T330 | 66 |
| FIGURE 1. 30 LENOVO THINK SYSTEM ST50 | 66 |
| FIGURE 1. 31 DELL POWEREDGE R440 SERVER..... | 67 |
| FIGURE 1. 32 HP PROLIANT DL380P G8 SERVER | 67 |
| FIGURE 1. 33 HP PROLIANT DL380 GEN10 | 67 |
| FIGURE 1. 34 DELL POWEREDGE M520 | 68 |
| FIGURE 1. 35 HP PROLIANT BL460C GEN8..... | 68 |
| FIGURE 1. 36 FUJITSU PY PRIMERGY BX924 S4 | 68 |
| FIGURE 1. 37 CISCO B200 M3 | 68 |
| FIGURE 1. 38 SERVER OPERATING SYSTEM TYPES | 70 |
| FIGURE 1. 39 WINDOWS SERVER 2016 INTERFACE | 72 |
| FIGURE 1. 40 LINUX SERVER OPERATING SYSTEM INTERFACE..... | 73 |
| FIGURE 1. 41 UNIX OPERATING SYSTEM INTERFACE..... | 74 |
| FIGURE 1. 42 NETWARE OPERATING SYSTEM INTERFACE | 75 |
| FIGURE 1. 43 RED HAT ENTERPRISE LINUX (RHEL) INTERFACE..... | 76 |
| FIGURE 1. 44 WALLPAPER OF WINDOWS SERVER 2016 | 77 |
| FIGURE 1. 45 MICROSOFT STORE ISN'T IN SERVER SOFTWARE..... | 81 |
| FIGURE 1. 46 SERVER HARDWARE..... | 82 |
| FIGURE 1. 47 WINDOWS10 OS DESKTOP DIFFERENCE WITH WINDOWS SERVER OS DESKTOP..... | 84 |

| | |
|---|-----|
| FIGURE 1. 48 WINDOWS10 OS WINDOWS KEY DIFFERENCE WITH WINDOWS SERVER OS DESKTOP | 85 |
| FIGURE 1. 49 PRICES FOR WINDOWS SERVER IN 2019 | 86 |
| FIGURE 1. 50 PRICES FOR WINDOWS 10 | 86 |
| FIGURE 1. 51 HYBRID SERVER..... | 89 |
| FIGURE 1. 52 HOW DNS BRIDGE THE COMMUNICATION GAP BETWEEN COMPUTERS AND HUMANS | 90 |
| FIGURE 1. 53 HOW DNS WORKS LIKE A PHONEBOOK | 91 |
| FIGURE 1. 54 PROCESS OF VIEWING A WEBSITE | 93 |
| FIGURE 1. 55 DNS HIERARCHY..... | 95 |
| FIGURE 1. 56 DOMAINS AND SUBDOMAINS IN DNS HIERARCHY | 96 |
| FIGURE 1. 57 RECURSIVE RESOLVER..... | 98 |
| FIGURE 1. 58 ROOT NAMESERVER | 99 |
| FIGURE 1. 59 TLD NAMESERVER | 100 |
| FIGURE 1. 60 AUTHORITATIVE NAMESERVER | 101 |
| FIGURE 1. 61 PARTS OF URL..... | 102 |
| FIGURE 1. 62 PARTS OF URL - SCHEME..... | 102 |
| FIGURE 1. 63 PARTS OF URL - SUBDOMAIN | 102 |
| FIGURE 1. 64 PART OF URL -SECOND LEVEL DOMAIN..... | 103 |
| FIGURE 1. 65 PARTS OF URL -TOP LEVEL DOMAIN..... | 103 |
| FIGURE 1. 66 PARTS OF URL -SUBDIRECTORY..... | 103 |
| FIGURE 1. 67 SEARCH ENGINE OPTIMIZATION (SEO) OPTIMIZATION..... | 104 |
| FIGURE 1. 68 SEARCH-RANKING-CLICK-THROUGH-RATES | 105 |
| FIGURE 1. 69 TOP 10 SEARCH ENGINES | 106 |
| FIGURE 1. 70 TESTING MY HMS SPEED BY “PANTHEON” WEBSITE SPEED TEST..... | 108 |
| FIGURE 1. 71 TESTING SPEED OF A MOBILE SITE BY “TEST MY SITE” | 109 |
| FIGURE 1. 72 ANGULAR FRAMEWORK..... | 116 |
| FIGURE 1. 73 REACT FRAMEWORK | 117 |
| FIGURE 1. 74 VUE FRAMEWORK | 118 |
| FIGURE 1. 75 EMBER FRAMEWORK | 119 |
| FIGURE 1. 76 BOOTSTRAP FRAMEWORK..... | 120 |
| FIGURE 1. 77 NODE.JS FRAMEWORK | 121 |
| FIGURE 1. 78 EXPRESS FRAMEWORK..... | 122 |
| FIGURE 1. 79 DJANGO FRAMEWORK..... | 123 |
| FIGURE 1. 80 RAILS FRAMEWORK | 124 |
| FIGURE 1. 81 WAMP SERVER’S LOCAL HOST | 127 |
| | |
| FIGURE 2. 1 RELATIONSHIPS BETWEEN FRONT-END AND BACK-END WEBSITE TECHNOLOGIES | 129 |
| FIGURE 2. 2 FRONT END DEVELOPMENT | 130 |
| FIGURE 2. 3 FRONT END DEVELOPING LANGUAGES | 131 |
| FIGURE 2. 4 BACK END DEVELOPMENT | 133 |
| FIGURE 2. 5 BACK-END PROGRAMMING LANGUAGES..... | 134 |
| FIGURE 2. 6 DIFFERENCE BETWEEN FRONTEND AND BACKEND | 136 |
| FIGURE 2. 7 FULL STACK DEVELOPMENT..... | 138 |
| FIGURE 2. 8 WEB APPLICATIONS' THREE CORE ARCHITECTURAL TIERS..... | 139 |
| FIGURE 2. 9 THEMEFORSET.NET HOME PAGE | 150 |
| FIGURE 2. 10 USING SUBLIME TEXT FOR CODE EDITING | 151 |
| FIGURE 2. 11 USING CHROME DEVELOPER TOOLS FOR DEVELOPING | 152 |
| FIGURE 2. 12 USING JQUERY-3.6.0.JS FOR DEVELOPMENT..... | 153 |
| FIGURE 2. 13 USING CODEPEN FOR DEVELOPING..... | 154 |
| FIGURE 2. 14 USING BOOTSTRAP FOR DEVELOPMENT | 155 |

| | |
|---|-----|
| FIGURE 3. 1 HOME PAGE WITH PURPOSE..... | 157 |
| FIGURE 3. 2 STAFF MANAGEMENT PAGE | 158 |
| FIGURE 3. 3 WARD MANAGEMENT PAGE | 158 |
| FIGURE 3. 4 WARD MANAGEMENT PAGE CODING WITH COMMENTS | 159 |
| FIGURE 3. 5 PATIENT MANAGEMENT PAGE BUTTONS WITH ICONS | 159 |
| FIGURE 3. 6 ABOUT US CODING PART WITH TYPEFACES..... | 160 |
| FIGURE 3. 7 ABOUT US CODING PART'S OUTPUT WITH TYPEFACES..... | 161 |
| FIGURE 3. 8 PATIENT MANAGEMENT PAGE WITH SUITABLE COLOURS..... | 161 |
| FIGURE 3. 9 HOME PAGE WITH CAROUSEL..... | 162 |
| FIGURE 3. 10 BLOG PAGE WITH PICTURES | 162 |
| FIGURE 3. 11 NAVIGATION BAR OF THE SYSTEM..... | 163 |
| FIGURE 3. 12 MEDICINES STOCK MANAGEMENT PAGE..... | 163 |
| FIGURE 3. 13 MEDICINES STOCK MANAGEMENT PAGE'S NAVIGATION BAR CODE PART | 164 |
| FIGURE 3. 14 MEDICINES STOCK MANAGEMENT PAGE'S TABLE CODE PART | 164 |
| FIGURE 3. 15 HOME PAGE DESKTOP MODE | 165 |
| FIGURE 3. 16 HOME PAGE MOBILE MODE | 166 |
| FIGURE 3. 17 ABOUT US PAGE DESKTOP MODE..... | 166 |
| FIGURE 3. 18 ABOUT US PAGE MOBILE MODE..... | 167 |
| FIGURE 3. 19 ER DIAGRAM FOR AROGYA HOSPITAL MANAGEMENT SYSTEM..... | 169 |
| FIGURE 3. 20 LOGICAL DATABASE DESIGN FOR AROGYA HOSPITAL MANAGEMENT SYSTEM..... | 170 |
| FIGURE 3. 21 CLASS DIAGRAM FOR AROGYA HOSPITAL MANAGEMENT SYSTEM..... | 172 |
| FIGURE 3. 22 USE CASE DIAGRAM FOR AROGYA HOSPITAL MANAGEMENT SYSTEM..... | 174 |
| FIGURE 3. 23 WIREFRAME SAMPLES | 188 |
| FIGURE 3. 24 LOGIN PAGE WIREFRAME | 189 |
| FIGURE 3. 25 HOME PAGE WIREFRAME | 189 |
| FIGURE 3. 26 ABOUT US PAGE WIREFRAME | 190 |
| FIGURE 3. 27 BLOG PAGE WIREFRAME | 190 |
| FIGURE 3. 28 CONTACT US PAGE WIREFRAME | 191 |
| FIGURE 3. 29 STAFF PAGE WIREFRAME..... | 191 |
| FIGURE 3. 30 PATIENTS PAGE WIREFRAME | 192 |
| FIGURE 3. 31 WARDS PAGE WIREFRAME | 192 |
| FIGURE 3. 32 ROOMS PAGE WIREFRAME | 193 |
| FIGURE 3. 33 MEDICINES PAGE WIREFRAME..... | 193 |
| FIGURE 3. 34 OTHER PAGE WIREFRAME | 194 |
| FIGURE 3. 35 PATIENT INVOICE PAGE WIREFRAME..... | 194 |
| FIGURE 3. 36 INITIAL OUTPUT OF LOGIN PAGE LOGIN PAGE..... | 195 |
| FIGURE 3. 37 INITIAL OUTPUT OF HOME PAGE | 195 |
| FIGURE 3. 38 INITIAL OUTPUT OF ABOUT US PAGE..... | 195 |
| FIGURE 3. 39 INITIAL OUTPUT OF BLOG PAGE | 196 |
| FIGURE 3. 40 INITIAL OUTPUT OF CONTACT US PAGE | 196 |
| FIGURE 3. 41 INITIAL OUTPUT OF STAFF PAGE | 196 |
| FIGURE 3. 42 INITIAL OUTPUT OF PATIENTS PAGE | 197 |
| FIGURE 3. 43 INITIAL OUTPUT OF WARDS PAGE | 197 |
| FIGURE 3. 44 INITIAL OUTPUT OF ROOMS PAGE | 197 |
| FIGURE 3. 45 INITIAL OUTPUT OF MEDICINES PAGE..... | 198 |
| FIGURE 3. 46 INITIAL OUTPUT OF OTHER PAGE..... | 198 |
| FIGURE 3. 47 MySQL DATABASE ("TEST" DATABASE)..... | 198 |
| FIGURE 3. 48 FILE ORGANIZATION IN WAMP SERVER PART 1 | 199 |
| FIGURE 3. 49 FILE ORGANIZATION IN WAMP SERVER PART 2 | 199 |
| FIGURE 3. 50 FINAL DESIGN OF INDEX.PHP..... | 200 |
| FIGURE 3. 51 CODING FOR INDEX.PHP PART 1 | 201 |

| | |
|--|-----|
| FIGURE 3. 52 CODING FOR INDEX.PHP PART 2 | 202 |
| FIGURE 3. 53 CODING FOR LOGIN_FUNCTION.PHP | 203 |
| FIGURE 3. 54 FINAL DESIGN OF HOME.PHP..... | 204 |
| FIGURE 3. 55 CODING FOR HOME.PHP PART 1..... | 205 |
| FIGURE 3. 56 CODING FOR HOME.PHP PART 2..... | 206 |
| FIGURE 3. 57 CODING FOR HOME.PHP PART 3..... | 207 |
| FIGURE 3. 58 CODING FOR HOME.PHP PART 4..... | 208 |
| FIGURE 3. 59 FINAL DESIGN OF ABOUT.PHP PART 1..... | 209 |
| FIGURE 3. 60 FINAL DESIGN OF ABOUT.PHP PART 2..... | 210 |
| FIGURE 3. 61 CODING FOR ABOUT.PHP PART 1..... | 211 |
| FIGURE 3. 62 CODING FOR ABOUT.PHP PART 2..... | 212 |
| FIGURE 3. 63 CODING FOR ABOUT.PHP PART 3..... | 213 |
| FIGURE 3. 64 CODING FOR ABOUT.PHP PART 4..... | 214 |
| FIGURE 3. 65 CODING FOR ABOUT.PHP PART 5..... | 215 |
| FIGURE 3. 66 FINAL DESIGN OF BLOG.PHP PART 1..... | 216 |
| FIGURE 3. 67 FINAL DESIGN OF BLOG.PHP PART 2..... | 217 |
| FIGURE 3. 68 CODING FOR BLOG.PHP PART 1..... | 218 |
| FIGURE 3. 69 CODING FOR BLOG.PHP PART 2..... | 219 |
| FIGURE 3. 70 CODING FOR BLOG.PHP PART 3 | 220 |
| FIGURE 3. 71 FINAL DESIGN OF CONTACT.PHP..... | 221 |
| FIGURE 3. 72 CODING FOR CONTACT.PHP PART 1..... | 222 |
| FIGURE 3. 73 CODING FOR CONTACT.PHP PART 2..... | 223 |
| FIGURE 3. 74 CODING FOR CONTACT.PHP PART 3..... | 224 |
| FIGURE 3. 75 CODING FOR CONTACT.PHP PART 4..... | 225 |
| FIGURE 3. 76 CODING FOR CONTACT.PHP PART 5..... | 226 |
| FIGURE 3. 77 FINAL DESIGN OF STAFF.PHP | 227 |
| FIGURE 3. 78 CODING FOR STAFF.PHP PART 1 | 228 |
| FIGURE 3. 79 CODING FOR STAFF.PHP PART 1 | 229 |
| FIGURE 3. 80 CODING FOR STAFF.PHP PART 2 | 230 |
| FIGURE 3. 81 FINAL DESIGN OF CREATE_STAFF.PHP | 231 |
| FIGURE 3. 82 CODING FOR CREATE_STAFF.PHP PART 1 | 232 |
| FIGURE 3. 83 CODING FOR CREATE_STAFF.PHP PART 2 | 233 |
| FIGURE 3. 84 CODING FOR CREATE_STAFF.PHP PART 3 | 234 |
| FIGURE 3. 85 FINAL DESIGN OF VIEW_STAFF.PHP | 235 |
| FIGURE 3. 86 CODING FOR VIEW_STAFF.PHP PART 1 | 236 |
| FIGURE 3. 87 CODING FOR VIEW_STAFF.PHP PART 2 | 237 |
| FIGURE 3. 88 FINAL DESIGN OF EDIT_STAFF.PHP | 238 |
| FIGURE 3. 89 CODING FOR EDIT_STAFF.PHP PART 1 | 239 |
| FIGURE 3. 90 CODING FOR EDIT_STAFF.PHP PART 2 | 240 |
| FIGURE 3. 91 CODING FOR EDIT_STAFF.PHP PART 3 | 241 |
| FIGURE 3. 92 CODING FOR DELETE_STAFF.PHP PART 1 | 242 |
| FIGURE 3. 93 FINAL DESIGN OF PATIENTS.PHP..... | 243 |
| FIGURE 3. 94 FINAL DESIGN OF PATIENT INVOICE MODAL | 244 |
| FIGURE 3. 95 CODING FOR PATIENTS.PHP PART 1 | 245 |
| FIGURE 3. 96 CODING FOR PATIENTS.PHP PART 2..... | 246 |
| FIGURE 3. 97 CODING FOR PATIENTS.PHP PART 3..... | 247 |
| FIGURE 3. 98 CODING FOR PATIENTS.PHP PART 4..... | 248 |
| FIGURE 3. 99 CODING FOR PATIENTS.PHP PART 5..... | 249 |
| FIGURE 3. 100 CODING FOR PATIENTS.PHP PART 6..... | 250 |
| FIGURE 3. 101 FINAL DESIGN OF CREATE_PATIENT.PHP | 251 |
| FIGURE 3. 102 CODING FOR CREATE_PATIENT.PHP PART 1..... | 252 |

| | |
|---|-----|
| FIGURE 3. 103 CODING FOR CREATE_PATIENT.PHP PART 2..... | 253 |
| FIGURE 3. 104 CODING FOR CREATE_PATIENT.PHP PART 3..... | 254 |
| FIGURE 3. 105 FINAL DESIGN OF VIEW_PATIENT.PHP..... | 255 |
| FIGURE 3. 106 CODING FOR VIEW_PATIENT.PHP PART 1..... | 256 |
| FIGURE 3. 107 CODING FOR VIEW_PATIENT.PHP PART 2..... | 257 |
| FIGURE 3. 108 FINAL DESIGN OF EDIT_PATIENT.PHP | 258 |
| FIGURE 3. 109 CODING FOR EDIT_PATIENT.PHP PART 1..... | 259 |
| FIGURE 3. 110 CODING FOR EDIT_PATIENT.PHP PART 2..... | 260 |
| FIGURE 3. 111 CODING FOR EDIT_PATIENT.PHP PART 3..... | 261 |
| FIGURE 3. 112 CODING FOR DELETE_PATIENT.PHP PART 1..... | 262 |
| FIGURE 3. 113 FINAL DESIGN OF WARDS.PHP | 263 |
| FIGURE 3. 114 CODING FOR WARDS.PHP PART 1 | 264 |
| FIGURE 3. 115 CODING FOR WARDS.PHP PART 2 | 265 |
| FIGURE 3. 116 CODING FOR WARDS.PHP PART 3 | 266 |
| FIGURE 3. 117 CODING FOR WARDS.PHP PART 4 | 267 |
| FIGURE 3. 118 FINAL DESIGN OF WARD_MANAGE.PHP | 268 |
| FIGURE 3. 119 CODING FOR WARD_MANAGE.PHP PART 1 | 269 |
| FIGURE 3. 120 CODING FOR WARD_MANAGE.PHP PART 2 | 270 |
| FIGURE 3. 121 CODING FOR WARD_MANAGE.PHP PART 3 | 271 |
| FIGURE 3. 122 CODING FOR WARD_MANAGE.PHP PART 4 | 272 |
| FIGURE 3. 123 FINAL DESIGN OF ROOMS.PHP | 273 |
| FIGURE 3. 124 CODING FOR ROOMS.PHP PART 1 | 274 |
| FIGURE 3. 125 CODING FOR ROOMS.PHP PART 2 | 275 |
| FIGURE 3. 126 CODING FOR ROOMS.PHP PART 3 | 276 |
| FIGURE 3. 127 CODING FOR ROOMS.PHP PART 4 | 277 |
| FIGURE 3. 128 FINAL DESIGN OF ROOM_MANAGE.PHP | 278 |
| FIGURE 3. 129 CODING FOR ROOM_MANAGE.PHP PART 1 | 279 |
| FIGURE 3. 130 CODING FOR ROOM_MANAGE.PHP PART 2 | 280 |
| FIGURE 3. 131 FINAL DESIGN OF MEDICINES.PHP | 281 |
| FIGURE 3. 132 CODING FOR MEDICINES.PHP PART 1 | 282 |
| FIGURE 3. 133 CODING FOR MEDICINES.PHP PART 2 | 283 |
| FIGURE 3. 134 CODING FOR MEDICINES.PHP PART 3 | 284 |
| FIGURE 3. 135 FINAL DESIGN OF CREATE_MEDICINE.PHP | 285 |
| FIGURE 3. 136 CODING FOR CREATE_MEDICINE.PHP PART 1 | 286 |
| FIGURE 3. 137 CODING FOR CREATE_MEDICINE.PHP PART 2 | 287 |
| FIGURE 3. 138 CODING FOR VIEW_MEDICINE.PHP PART 1 | 288 |
| FIGURE 3. 139 CODING FOR VIEW_MEDICINE.PHP PART 2 | 289 |
| FIGURE 3. 140 FINAL DESIGN OF EDIT_MEDICINE.PHP..... | 290 |
| FIGURE 3. 141 CODING FOR EDIT_MEDICINE.PHP PART 1 | 291 |
| FIGURE 3. 142 CODING FOR EDIT_MEDICINE.PHP PART 2 | 292 |
| FIGURE 3. 143 CODING FOR DELETE_MEDICINE.PHP PART 1 | 293 |
| FIGURE 3. 144 FINAL DESIGN OF OTHER.PHP | 294 |
| FIGURE 3. 145 CODING FOR OTHER.PHP PART 1 | 295 |
| FIGURE 3. 146 CODING FOR OTHER.PHP PART 2 | 296 |
| FIGURE 3. 147 CODING FOR OTHER.PHP PART 3 | 297 |
| FIGURE 3. 148 FINAL DESIGN OF CREATE_OTHER.PHP | 298 |
| FIGURE 3. 149 CODING FOR CREATE_OTHER.PHP PART 1 | 299 |
| FIGURE 3. 150 CODING FOR CREATE_OTHER.PHP PART 2 | 300 |
| FIGURE 3. 151 FINAL DESIGN OF VIEW_OTHER.PHP | 301 |
| FIGURE 3. 152 CODING FOR VIEW_OTHER.PHP PART 1..... | 302 |
| FIGURE 3. 153 CODING FOR VIEW_OTHER.PHP PART 2..... | 303 |

| | |
|---|-----|
| FIGURE 3. 154 FINAL DESIGN OF EDIT_OTHER.PHP | 304 |
| FIGURE 3. 155 CODING FOR EDIT_OTHER.PHP PART 1 | 305 |
| FIGURE 3. 156 CODING FOR EDIT_OTHER.PHP PART 2 | 306 |
| FIGURE 3. 157 CODING FOR DB.PHP | 307 |
| FIGURE 3. 158 CODING FOR HEADER.PHP | 307 |
| FIGURE 3. 159 CODING FOR FOOTER.PHP | 308 |
| FIGURE 3. 160 CODING FOR MYSTYLE.CSS | 308 |
| FIGURE 3. 161 FINAL PRODUCT'S FILE ORGANIZATION IN WAMP SERVER PART 1 | 309 |
| FIGURE 3. 162 FINAL PRODUCT'S FILE ORGANIZATION IN WAMP SERVER PART 2 | 309 |
| FIGURE 3. 163 FINAL PRODUCT'S FILE ORGANIZATION IN WAMP SERVER PART 3 | 310 |
| FIGURE 3. 164 FINAL PRODUCT'S FILE ORGANIZATION IN WAMP SERVER PART 4 | 310 |
| FIGURE 3. 165 FINAL PRODUCT'S FILE ORGANIZATION IN WAMP SERVER PART 5 | 311 |
| FIGURE 3. 166 FINAL PRODUCT'S FILE ORGANIZATION IN WAMP SERVER PART 6 | 311 |
| FIGURE 3. 167 FINAL PRODUCT'S FILE ORGANIZATION IN WAMP SERVER PART 7 | 312 |
| FIGURE 3. 168 FINAL PRODUCT'S FILE ORGANIZATION IN WAMP SERVER PART 8 | 312 |
| FIGURE 3. 169 FINAL PRODUCT'S FILE ORGANIZATION IN WAMP SERVER PART 9 | 313 |
| FIGURE 3. 170 FINAL PRODUCT'S FILE ORGANIZATION IN WAMP SERVER PART 10 | 313 |
| FIGURE 3. 171 FINAL PRODUCT'S FILE ORGANIZATION IN WAMP SERVER PART 11 | 314 |
| FIGURE 3. 172 FINAL PRODUCT'S FILE ORGANIZATION IN WAMP SERVER PART 12 | 314 |
| FIGURE 3. 173 FINAL PRODUCT'S DATABASE INTERFACES [MYSQL DATABASE ("PHP_CRUD" DATABASE)] | 315 |
| FIGURE 3. 174 FINAL PRODUCT'S DATABASE INTERFACES [ADMIN TABLE] – TABLE STRUCTURE | 316 |
| FIGURE 3. 175 FINAL PRODUCT'S DATABASE INTERFACES [ADMIN TABLE] – INSERTED DATA | 316 |
| FIGURE 3. 176 FINAL PRODUCT'S DATABASE INTERFACES [BILLS TABLE] – TABLE STRUCTURE | 317 |
| FIGURE 3. 177 FINAL PRODUCT'S DATABASE INTERFACES [BILLS TABLE] – INSERTED DATA | 317 |
| FIGURE 3. 178 FINAL PRODUCT'S DATABASE INTERFACES [MEDICINE TABLE] – TABLE STRUCTURE | 318 |
| FIGURE 3. 179 FINAL PRODUCT'S DATABASE INTERFACES [MEDICINE TABLE] – INSERTED DATA | 318 |
| FIGURE 3. 180 FINAL PRODUCT'S DATABASE INTERFACES [OTHER TABLE] – TABLE STRUCTURE | 319 |
| FIGURE 3. 181 FINAL PRODUCT'S DATABASE INTERFACES [OTHER TABLE] – INSERTED DATA | 319 |
| FIGURE 3. 182 FINAL PRODUCT'S DATABASE INTERFACES [PATIENT TABLE] – TABLE STRUCTURE | 320 |
| FIGURE 3. 183 FINAL PRODUCT'S DATABASE INTERFACES [PATIENT TABLE] – INSERTED DATA | 320 |
| FIGURE 3. 184 FINAL PRODUCT'S DATABASE INTERFACES [ROOMS TABLE] – TABLE STRUCTURE | 321 |
| FIGURE 3. 185 FINAL PRODUCT'S DATABASE INTERFACES [ROOMS TABLE] – INSERTED DATA | 321 |
| FIGURE 3. 186 FINAL PRODUCT'S DATABASE INTERFACES [STAFF TABLE] – TABLE STRUCTURE | 322 |
| FIGURE 3. 187 FINAL PRODUCT'S DATABASE INTERFACES [STAFF TABLE] – INSERTED DATA | 322 |
| FIGURE 3. 188 FINAL PRODUCT'S DATABASE INTERFACES [WARDS TABLE] – TABLE STRUCTURE | 323 |
| FIGURE 3. 189 FINAL PRODUCT'S DATABASE INTERFACES [WARDS TABLE] – INSERTED DATA | 323 |
| FIGURE 3. 190 FINAL PRODUCT'S DATABASE CODES PART 1 | 324 |
| FIGURE 3. 191 FINAL PRODUCT'S DATABASE CODES PART 2 | 325 |
| FIGURE 3. 192 FINAL PRODUCT'S DATABASE CODES PART 3 | 326 |
| FIGURE 3. 193 FINAL PRODUCT'S DATABASE CODES PART 4 | 327 |
| FIGURE 3. 194 FINAL PRODUCT'S DATABASE CODES PART 5 | 328 |
| FIGURE 3. 195 FINAL PRODUCT'S DATABASE CODES PART 6 | 328 |
| FIGURE 3. 196 LOGIN PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE DESIGNED DOCUMENT(WIREFRAME) | 332 |
| FIGURE 3. 197 HOME PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE DESIGNED DOCUMENT(WIREFRAME) | 334 |
| FIGURE 3. 198 ABOUT US PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE DESIGNED DOCUMENT(WIREFRAME) | 337 |
| FIGURE 3. 199 BLOG PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE DESIGNED DOCUMENT(WIREFRAME) | 340 |

| | |
|--|-----|
| FIGURE 3. 200 CONTACT US PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE DESIGNED DOCUMENT(WIREFRAME)..... | 342 |
| FIGURE 3. 201 STAFF PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE DESIGNED DOCUMENT(WIREFRAME)..... | 344 |
| FIGURE 3. 202 PATIENTS PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE DESIGNED DOCUMENT(WIREFRAME)..... | 346 |
| FIGURE 3. 203 WARDS PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE DESIGNED DOCUMENT(WIREFRAME)..... | 348 |
| FIGURE 3. 204 ROOMS PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE DESIGNED DOCUMENT(WIREFRAME)..... | 350 |
| FIGURE 3. 205 MEDICINES PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE DESIGNED DOCUMENT(WIREFRAME)..... | 352 |
| FIGURE 3. 206 OTHER PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE DESIGNED DOCUMENT(WIREFRAME)..... | 354 |
| FIGURE 3. 207 INVOICE MODAL DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE DESIGNED DOCUMENT(WIREFRAME)..... | 356 |
| FIGURE 3. 208 LOGIN PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE INITIAL OUTPUT. | 358 |
| FIGURE 3. 209 HOME PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE INITIAL OUTPUT | 360 |
| FIGURE 3. 210 ABOUT US PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE INITIAL OUTPUT | 363 |
| FIGURE 3. 211 BLOG PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE INITIAL OUTPUT.. | 366 |
| FIGURE 3. 212 CONTACT US PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE INITIAL OUTPUT | 368 |
| FIGURE 3. 213 STAFF PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE INITIAL OUTPUT. | 370 |
| FIGURE 3. 214 PATIENTS PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE INITIAL OUTPUT | 372 |
| FIGURE 3. 215 WARDS PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE INITIAL OUTPUT | 374 |
| FIGURE 3. 216 ROOMS PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE INITIAL OUTPUT | 376 |
| FIGURE 3. 217 MEDICINES PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE INITIAL OUTPUT | 378 |
| FIGURE 3. 218 OTHER PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE INITIAL OUTPUT | 380 |
| | |
| FIGURE 4. 1 LOGIN PAGE INTERFACE | 395 |
| FIGURE 4. 2 HOME PAGE INTERFACE..... | 396 |
| FIGURE 4. 3 ABOUT Us PAGE INTERFACE | 397 |
| FIGURE 4. 4 BLOG PAGE INTERFACE | 398 |
| FIGURE 4. 5 CONTACT Us PAGE INTERFACE | 399 |
| FIGURE 4. 6 STAFF MANAGEMENT PAGE INTERFACE | 400 |
| FIGURE 4. 7 PATIENT MANAGEMENT PAGE INTERFACE | 401 |
| FIGURE 4. 8 WARDS MANAGEMENT PAGE INTERFACE | 402 |
| FIGURE 4. 9 ROOMS MANAGEMENT PAGE INTERFACE | 403 |
| FIGURE 4. 10 MEDICINE STOCK MANAGEMENT PAGE INTERFACE | 404 |
| FIGURE 4. 11 OTHER SUPPLIES MANAGEMENT PAGE INTERFACE | 405 |
| FIGURE 4. 12 USER FEEDBACK FORM PART 1 | 448 |
| FIGURE 4. 13 USER FEEDBACK FORM PART 2 | 449 |
| FIGURE 4. 14 USER FEEDBACK FORM PART 3 | 450 |
| FIGURE 4. 15 USER FEEDBACK FORM PART 4 | 451 |
| FIGURE 4. 16 USING SIMPLE IMAGES FOR CUSTOMER RATING SCORES..... | 452 |

| | |
|---|-----|
| FIGURE 4. 17 SIDE FEEDBACK BUTTON AT THE EDGE OF THE SYSTEM | 453 |
| FIGURE 4. 18 INSTANT FEEDBACK SAMPLE FROM COOPERATE WEBSITE | 454 |

List of Tables

| | |
|---|-----|
| TABLE 1. 1 DYNAMIC VS STATIC WEB SERVERS | 44 |
| TABLE 1. 2 HP PROLIANT PRODUCT LINES WITH THEIR FORM FACTOR | 58 |
| TABLE 1. 3 HP PROLIANT DIFFERENT SERIES WITH NUMBER OF CPU SOCKETS | 59 |
| TABLE 1. 4 DL 380 DIFFERENCE BETWEEN GENERATION 1 AND GENERATION 10 | 60 |
| TABLE 1. 5 DELL POWEREDGE FIRST DIGIT VS THE NUMBER OF CPU SOCKETS..... | 61 |
| TABLE 1. 6 PROS AND CONS OF BUYING CHEAPER SERVER AND EXPENSIVE SERVER..... | 63 |
| TABLE 1. 7 ADVANTAGES AND DISADVANTAGES OF HAVING ON-SITE SERVER | 64 |
| TABLE 1. 8 ADVANTAGES AND DISADVANTAGES OF HAVING CLOUD SERVER..... | 65 |
| TABLE 1. 9 CLOUD SERVERS PRICE BY "SERVER MANIA" | 69 |
| TABLE 1. 10 HYBRID SERVERS PRICE BY "SERVER MANIA" | 69 |
| TABLE 1. 11 WINDOWS 10 HARDWARE REQUIREMENTS AGAINST WITH WINDOWS SERVER HARDWARE REQUIREMENTS | 83 |
| TABLE 1. 12 DIFFERENCES BETWEEN ON-SITE SERVERS AND CLOUD SERVERS | 88 |
| TABLE 1. 13 TOP-LEVEL DNS DOMAINS WITH EXAMPLES | 94 |
| TABLE 2. 1 DIFFERENCE BETWEEN FRONTEND AND BACKEND..... | 137 |
| TABLE 2. 2 DIFFERENCES BETWEEN ONLINE WEBSITE CREATION TOOLS AND CUSTOM-BUILT WEBSITES | 147 |
| TABLE 3. 1 USER REQUIREMENT AND EFFECTIVENESS OF DESIGNED FORMS TO FULFILL THE USER REQUIREMENT..... | 180 |
| TABLE 3. 2 SYSTEM REQUIREMENT AND EFFECTIVENESS OF SOLUTION TO FULFILL THE SYSTEM REQUIREMENT..... | 184 |
| TABLE 3. 3 OVERALL DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE DESIGNED DOCUMENT(WIREFRAME)..... | 330 |
| TABLE 3. 4 LOGIN PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE DESIGNED DOCUMENT(WIREFRAME)..... | 331 |
| TABLE 3. 5 HOME PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE DESIGNED DOCUMENT(WIREFRAME)..... | 333 |
| TABLE 3. 6 ABOUT US PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE DESIGNED DOCUMENT(WIREFRAME)..... | 335 |
| TABLE 3. 7 BLOG PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE DESIGNED DOCUMENT(WIREFRAME)..... | 338 |
| TABLE 3. 8 CONTACT US PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE DESIGNED DOCUMENT(WIREFRAME)..... | 341 |
| TABLE 3. 9 STAFF PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE DESIGNED DOCUMENT(WIREFRAME)..... | 343 |
| TABLE 3. 10 PATIENTS PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE DESIGNED DOCUMENT(WIREFRAME)..... | 345 |
| TABLE 3. 11 WARDS PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE DESIGNED DOCUMENT(WIREFRAME)..... | 347 |
| TABLE 3. 12 ROOMS PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE DESIGNED DOCUMENT(WIREFRAME)..... | 349 |
| TABLE 3. 13 MEDICINES PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE DESIGNED DOCUMENT(WIREFRAME)..... | 351 |
| TABLE 3. 14 OTHER PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE DESIGNED DOCUMENT(WIREFRAME)..... | 353 |
| TABLE 3. 15 INVOICE MODAL DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE DESIGNED DOCUMENT(WIREFRAME)..... | 355 |

| | |
|---|-----|
| TABLE 3. 16 LOGIN PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE INITIAL OUTPUT | 357 |
| TABLE 3. 17 HOME PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE INITIAL OUTPUT | 359 |
| TABLE 3. 18 ABOUT US PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE INITIAL OUTPUT | 361 |
| TABLE 3. 19 BLOG PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE INITIAL OUTPUT | 364 |
| TABLE 3. 20 CONTACT US PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE INITIAL OUTPUT | 367 |
| TABLE 3. 21 STAFF PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE INITIAL OUTPUT..... | 369 |
| TABLE 3. 22 PATIENTS PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE INITIAL OUTPUT | 371 |
| TABLE 3. 23 WARDS PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE INITIAL OUTPUT .. | 373 |
| TABLE 3. 24 ROOMS PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE INITIAL OUTPUT .. | 375 |
| TABLE 3. 25 MEDICINES PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE INITIAL OUTPUT | 377 |
| TABLE 3. 26 OTHER PAGE DIFFERENCE BETWEEN MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE INITIAL OUTPUT ... | 379 |
| TABLE 4. 1 TEST PLAN FOR THE AROGYA HOSPITAL MANAGEMENT SYSTEM | 407 |
| TABLE 4. 2 TEST CASE 1 | 412 |
| TABLE 4. 3 TEST CASE 2 | 413 |
| TABLE 4. 4 TEST CASE 3 | 414 |
| TABLE 4. 5 TEST CASE 4 | 415 |
| TABLE 4. 6 TEST CASE 5..... | 416 |
| TABLE 4. 7 TEST CASE 6 | 417 |
| TABLE 4. 8 TEST CASE 7 | 418 |
| TABLE 4. 9 TEST CASE 8 | 419 |
| TABLE 4. 10 TEST CASE 9 | 420 |
| TABLE 4. 11 TEST CASE 10 | 421 |
| TABLE 4. 12 TEST CASE 11 | 422 |
| TABLE 4. 13 TEST CASE 12 | 423 |
| TABLE 4. 14 TEST CASE 13 | 424 |
| TABLE 4. 15 TEST CASE 14 | 425 |
| TABLE 4. 16 TEST CASE 15 | 426 |
| TABLE 4. 17 TEST CASE 16 | 427 |
| TABLE 4. 18 TEST CASE 17 | 428 |
| TABLE 4. 19 TEST CASE 18 | 429 |
| TABLE 4. 20 TEST CASE 19 | 430 |
| TABLE 4. 21 TEST CASE 20 | 431 |
| TABLE 4. 22 TEST CASE 21 | 432 |
| TABLE 4. 23 TEST CASE 22 | 433 |
| TABLE 4. 24 TEST CASE 23 | 434 |
| TABLE 4. 25 TEST CASE 24 | 435 |
| TABLE 4. 26 TEST CASE 25 | 436 |
| TABLE 4. 27 TEST CASE 26 | 437 |
| TABLE 4. 28 EVALUATION RESULTS OF THE TEST PLAN..... | 438 |

TABLE OF CONTENTS

| | |
|---|-----------|
| TASK 1..... | 31 |
| 1.1 PROTOCOLS | 31 |
| 1.1.1 <i>Introduction</i> | 31 |
| 1.1.2 <i>Purpose and Relationships between Communication Protocols</i> | 32 |
| 1.1.3 <i>Types of communication protocols</i> | 34 |
| 1.2 WEB SERVERS..... | 41 |
| 1.2.1 <i>Introduction</i> | 41 |
| 1.2.2 <i>What is Web Server</i> | 42 |
| 1.2.3 <i>Web Server Process</i> | 43 |
| 1.2.4 <i>Dynamic vs Static web servers</i> | 44 |
| 1.2.5 <i>Web server examples</i> | 46 |
| 1.3 SERVERS..... | 49 |
| 1.3.1 <i>Introduction</i> | 49 |
| 1.3.2 <i>Main 3 Types of Servers</i> | 50 |
| 1.3.3 <i>Other Types of Servers</i> | 54 |
| 1.3.4 <i>Server Clusters</i> | 55 |
| 1.4 SERVER HARDWARE TYPES..... | 56 |
| 1.4.1 <i>Introduction</i> | 56 |
| 1.4.2 <i>How to understand Server Hardware Name and Brands</i> | 58 |
| 1.4.3 <i>Things to know before buy a Server Hardware</i> | 62 |
| 1.4.4 <i>Other things to consider when buying a Server Hardware</i> | 63 |
| 1.4.5 <i>Server Hardware Prices</i> | 66 |
| 1.5 SERVER OPERATING SYSTEM | 70 |
| 1.5.1 <i>Introduction</i> | 70 |
| 1.5.2 <i>Server Operating System Features</i> | 71 |
| 1.5.3 <i>Server Operating System Benefits</i> | 71 |
| 1.5.4 <i>Server Operating System Types</i> | 72 |
| 1.6 WINDOWS SERVER..... | 77 |
| 1.6.1 <i>Introduction of Windows Server</i> | 77 |
| 1.6.2 <i>Operating System similarities between Windows and Windows Server</i> | 78 |
| 1.6.3 <i>Operating System differences between Windows and Windows Server</i> | 79 |
| 1.6.4 <i>Hardware differences between Desktop PC and Server</i> | 82 |
| 1.6.5 <i>Environment differences between Desktop PC and Server</i> | 84 |
| 1.6.6 <i>Server options limitations</i> | 85 |
| 1.6.7 <i>Price differences between Windows and Windows Server</i> | 86 |
| 1.6.8 <i>Differences between on-site servers and cloud servers</i> | 88 |
| 1.7 CHOOSING A SERVER TYPE FOR HOSPITAL MANAGEMENT SYSTEM..... | 89 |
| 1.8 DNS AND DNS SERVER | 90 |
| 1.8.1 <i>Introduction</i> | 90 |
| 1.8.2 <i>How DNS works</i> | 91 |
| 1.8.3 <i>Process of viewing a Website</i> | 92 |
| 1.9 DNS HIERARCHY | 94 |
| 1.9.1 <i>DNS Hierarchy with Top Level Domains</i> | 94 |
| 1.9.2 <i>Domains and Subdomains in DNS Hierarchy</i> | 95 |
| 1.10 DNS TYPES..... | 97 |

| | |
|---|----------------|
| 1.10.1 <i>Types of DNS and uses of them</i> | 97 |
| 1.11 WHAT IS URL AND PARTS OF URL | 102 |
| 1.12 SEARCH ENGINE OPTIMIZATION (SEO) OPTIMIZATION | 104 |
| 1.12.1 <i>Introduction</i> | 104 |
| 1.12.2 <i>Search Engine rankings</i> | 105 |
| 1.12.3 <i>Improving a site's index value and rank of the Hospital Management System (HMS) through search engine optimization</i> | 106 |
| 1.13 SEO TOOLS TO USE..... | 110 |
| 1.14 WEB DEVELOPMENT TECHNOLOGIES | 111 |
| 1.14.1 <i>Introduction</i> | 111 |
| 1.14.2 <i>Server-side (back-end) technology</i> | 113 |
| 1.15 WEB DEVELOPMENT FRAMEWORKS | 115 |
| 1.15.1 <i>Introduction</i> | 115 |
| 1.15.2 <i>Client-side (front-end) Framework</i> | 116 |
| 1.15.3 <i>Server-side (back-end) Framework</i> | 121 |
| 1.16 TOOLS AND TECHNIQUES CHOSEN TO DEVELOP THE HMS WEB APPLICATION..... | 125 |
| TASK 2..... | 129 |
| 2.1 RELATIONSHIPS BETWEEN FRONT-END AND BACK-END WEBSITE TECHNOLOGIES..... | 129 |
| 2.1.1 <i>Introduction</i> | 129 |
| 2.1.2 <i>What is Front End Development</i> | 130 |
| 2.1.3 <i>Front-End Programming Languages</i> | 131 |
| 2.1.4 <i>Frameworks and libraries for Front-End Development</i> | 132 |
| 2.1.5 <i>What is Back End Development</i> | 133 |
| 2.1.6 <i>Back-End Programming Languages</i> | 134 |
| 2.1.7 <i>Frameworks for Back-End Development</i> | 135 |
| 2.2 DIFFERENCE BETWEEN FRONTEND AND BACKEND | 136 |
| 2.3 FULL STACK DEVELOPMENT | 138 |
| 2.4 THE THREE-LAYER ARCHITECTURE | 139 |
| 2.4.1 <i>What is three-layer architecture</i> | 139 |
| 2.4.2 <i>Front-End and Back-End web development relationship with 3 layer architecture's Presentation layer, Application layer and Data layer</i> | 140 |
| 2.4.3 <i>Benefits of three-layer architecture</i> | 141 |
| 2.4.5 <i>Three-layer architecture in web development</i> | 142 |
| 2.5 ONLINE WEBSITE CREATION TOOLS AND CUSTOM-BUILT WEBSITES | 143 |
| 2.5.1 <i>Online Web Creation Tools</i> | 143 |
| 2.5.2 <i>Custom-Built Websites</i> | 145 |
| 2.6 TOOLS AND TECHNIQUES USED TO DESIGN THE HOSPITAL MANAGEMENT SYSTEM. | 150 |
| TASK 3..... | 157 |
| 3.1 HOSPITAL MANAGEMENT SYSTEM DESIGN PLANNING..... | 157 |
| 3.1.1 <i>Design principles to create the website</i> | 157 |
| 3.1.2 <i>ERD with Logical Database design for the Arogya Hospital Management System</i> | 168 |
| 3.1.3 <i>Class Diagram for the Arogya Hospital Management System</i> | 171 |
| 3.1.4 <i>Use Case Diagram for the Arogya Hospital Management System</i> | 173 |
| 3.2 TYPES OF REQUIREMENTS IN SYSTEM DEVELOPMENT | 175 |
| 3.2.1 <i>Introduction</i> | 175 |
| 3.2.2 <i>Business Requirements</i> | 175 |

| | |
|--|----------------|
| 3.2.3 Functional Requirements | 176 |
| 3.2.4 Quality Requirements..... | 176 |
| 3.2.5 User Requirements..... | 177 |
| 3.2.6 System Requirements..... | 178 |
| 3.2.7 Evaluate the effectiveness of designed forms with identified user requirements..... | 180 |
| 3.2.8 Evaluate the effectiveness of designed forms with identified system requirements | 184 |
| 3.3 HOSPITAL MANAGEMENT SYSTEM OUTPUTS | 188 |
| 3.3.1 Initial Design (Wireframes) | 188 |
| 3.3.2 Initial Output..... | 195 |
| 3.3.3 Final Design with Codes (Final Output) | 200 |
| 3.3.4 Final product's File Organization in WAMP Server | 309 |
| 3.3.5 Final product's Database interfaces..... | 315 |
| 3.3.6 Final product's Database codes | 324 |
| 3.4 COMPARE AND CONTRAST THE MULTIPAGE WEBSITE CREATED (FINAL OUTPUT) TO THE DESIGNED DOCUMENT(WIREFRAME)..... | 329 |
| 3.4.1 Overall Difference | 329 |
| 3.4.2 Compare and Contrast each final output's pages with the initial design wireframe | 331 |
| 3.4.3 Compare and Contrast each final output's pages with the initial output | 357 |
| 3.4.4 Technical challenges faced during the development. | 381 |
| TASK 4..... | 384 |
| 4.1 THE IMPORTANCE OF QUALITY ASSURANCE IN THE WEB DEVELOPMENT PROCESS..... | 384 |
| 4.1.1 Introduction..... | 384 |
| 4.1.2 Types of QA Testing..... | 386 |
| 4.1.3 Tools to use in QA..... | 389 |
| 4.1.4 How QA helped to build the Hospital Management System | 391 |
| 4.2 DOCUMENTATION FOR DEVELOPED SYSTEM | 392 |
| 4.2.1 Why system documentation | 392 |
| 4.3 USER DOCUMENTATION..... | 393 |
| 4.3.1 What is user documentation | 393 |
| 4.3.2 Types of User Documentation | 393 |
| 4.3.3 User Documentation for Arogya Hospital Management System | 394 |
| 4.4 TEST PLAN | 406 |
| 4.4.1 Introduction..... | 406 |
| 4.4.2 Test Plan for the Arogya Hospital Management System | 407 |
| 4.5 TEST CASES..... | 411 |
| 4.5.1 Introduction..... | 411 |
| 4.5.2 Testing Cases according to Test Plan for the Arogya Hospital Management System | 412 |
| 4.6 EVALUATION RESULTS OF THE TEST PLAN | 438 |
| 4.7 TYPES OF SOFTWARE TESTING | 443 |
| 4.7.1 Introduction..... | 443 |
| 4.7.2 Testing Types..... | 444 |
| 4.8 FEEDBACK FORM | 448 |
| 4.8.1 Feedback form of sample survey..... | 448 |
| 4.8.2 How I can improve the feedback collection form | 452 |
| 4.9 FUTURE ENHANCEMENTS | 455 |
| CONCLUSION | 457 |
| REFERENCES..... | 458 |

Task 1

1.1 Protocols

1.1.1 Introduction

A protocol is a set of rules for formatting and processing data in networking. Computer protocols are similar to a common language. Although the software and hardware used by the computers in a network may differ greatly, protocols allow them to communicate with one another.

If one computer and another computer both utilize the Internet Protocol (IP), they will be able to communicate in the same way as the United Nations uses its six official languages to speak with delegates from all over the world. However, if one computer uses IP and the other does not, they will not be able to interact. There are many protocols for different types of procedures on the Internet. Protocols are frequently categorized according to whatever OSI model layer they belong to.

1.1.2 Purpose and Relationships between Communication Protocols

Communication protocols are formally defined standards and formats for digital messages. They must communicate within or between computing systems. In telecommunications networks and other systems, communication protocols are crucial because they establish uniformity and consistency for message transmission and receiving.

Authentication, error detection, and signaling can all be covered by communications protocols. Additionally, they can explain the synchronization of analog and digital communications as well as their syntax and semantics. Hardware and software both implement communication protocols. Everywhere in analog and digital communications, there are thousands of different communication protocols in use. Without them, computer networks would not be possible.

A protocol can specify many different aspects of a transmission. Properties that may be addressed by protocols include:

- Packet size.
- Speed of transmission.
- Types of error correction.
- Synchronization mechanisms and handshakes.
- Mapping of addresses.
- Recognition procedures.
- Flow management.
- Packet sequence management.
- Routing.
- Format for addresses.

Popular protocols include:

- File Transfer Protocol (FTP)
- TCP/IP
- User Datagram Protocol (UDP)
- Hypertext Transfer Protocol (HTTP)
- Post Office Protocol (POP3)
- Internet Message Access Protocol (IMAP)
- Simple Mail Transfer Protocol (SMTP).

Communications protocols provide rules in a digital environment that is mostly unregulated. On the Internet, for instance, the communication protocols developed by organizations like the World Wide Web Consortium (W3C) and the Internet engineering task force (IETF) contribute to providing universal operations and limiting various types of liability and vulnerabilities in these technologies. Protocols for authentication are a good example. The authentication protocols protect against typical techniques used by malicious hackers to compromise or muddle network communication, such as distributed denial of service (DDoS) assaults and other common methods.

When HTTP is changed to HTTPS and the widespread usage of SSL certificates becomes the standard. This is the intention of people who develop such communication protocols. The protocols function fundamentally to "harden" systems against various types of misuse, which is a significant element of their utility. Other communications protocols govern the utilization of data packets in global network pathways, which can resemble particle physics at times. It's not like an open tap, where information flows freely from one location to another.

The message between the sender and receiver is made up of small, distinct units, with each data packet having its own header, core data, and routing path. Communications protocols play such a strong and crucial role in this because everything needs to be carefully coordinated and planned. Additionally, the underlying technology itself affects the communication protocols that are employed. Consequently, protocols also advance along with technology. Because of this, Internet protocols will continue to improve and we'll probably witness very different communication protocols in tomorrow's developing networks.

1.1.3 Types of communication protocols

1) HTTP Protocol

The HTTP stands for Hyper Text Transfer Protocol. This protocol is the most widely used in the world. On the Internet, the HTTP protocol is used to access web pages. When we type a web address into a browser, HTTP is automatically added to the beginning of the address. This signifies that we are currently retrieving this web page over the HTTP protocol. HTTP transmits all data in plain text. As a result, there isn't any encryption. If we send in important information like passwords or credit card numbers, this would be a major problem. This is when HTTPS comes into play.

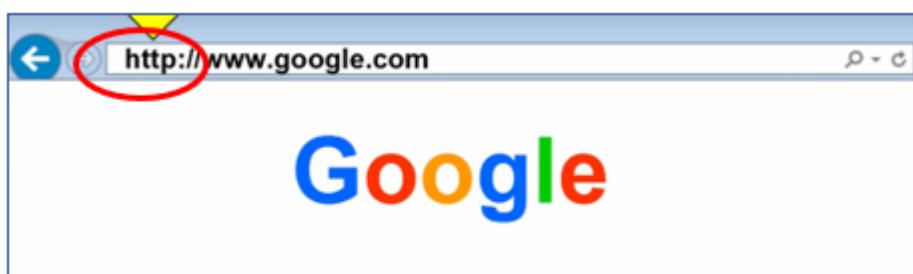


Figure 1. 1 Using HTTP Protocol

2) HTTPS Protocol

The HTTPS stands for Secure Hyper Text Transfer Protocol.

The HTTPS protocol is used to browse web pages on the Internet while ensuring that they are encrypted.

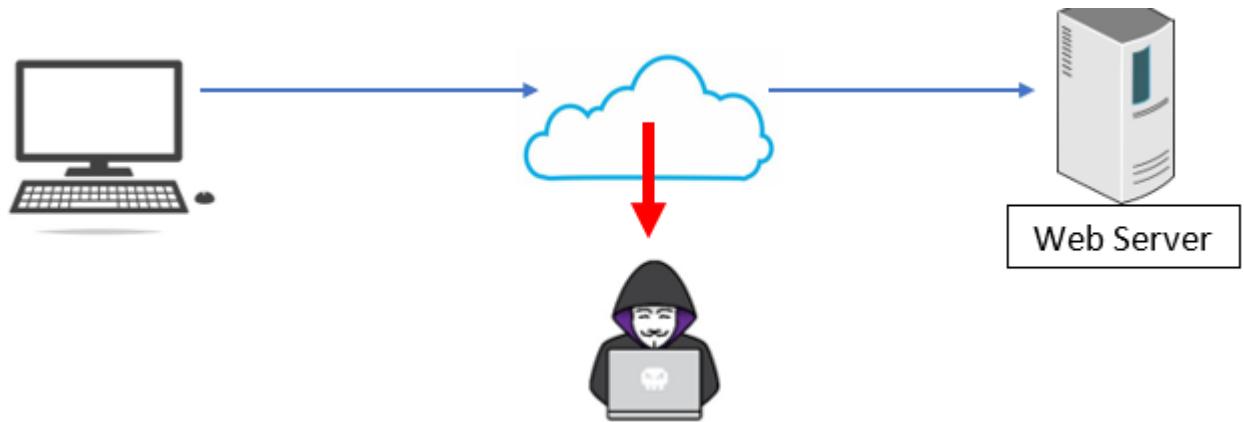


Figure 1. 2 Hacker tries to steal data through internet

Because of this protocol, if a hacker tries to steal data, he will get meaningless data.

One of these two protocols is used for HTTPS.

- 1) SSL (Secure Sockets Layer)
- 2) TLS (Transport Layer Security)

SSL →

The SSL protocol is used to guarantee internet security. To protect data, Public Key Encryption is used. When a computer connects to a website via SSL, it will request that the website identify itself. The web server will then deliver a copy of its SSL certificate to the computer. This certificate is a digital certificate that is used to verify a website's authenticity. SSL is a security protocol that informs our computer that the website we are viewing is secure.

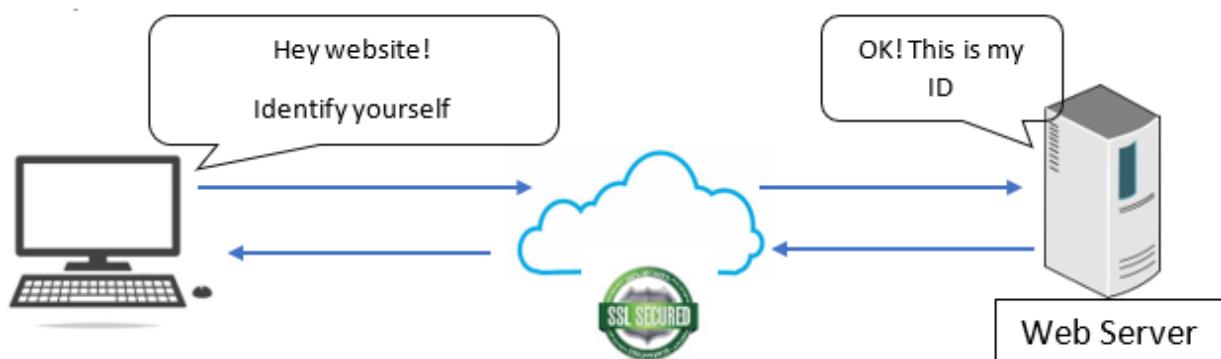


Figure 1. 3 How SSL acts when internet browsing

TCL →

This is SSL's successor, and it is based on the same specification. It encrypts data similarly to SSL. Many websites now use HTTPS as their default protocol. Because websites that aren't secure are now flagged by Google.

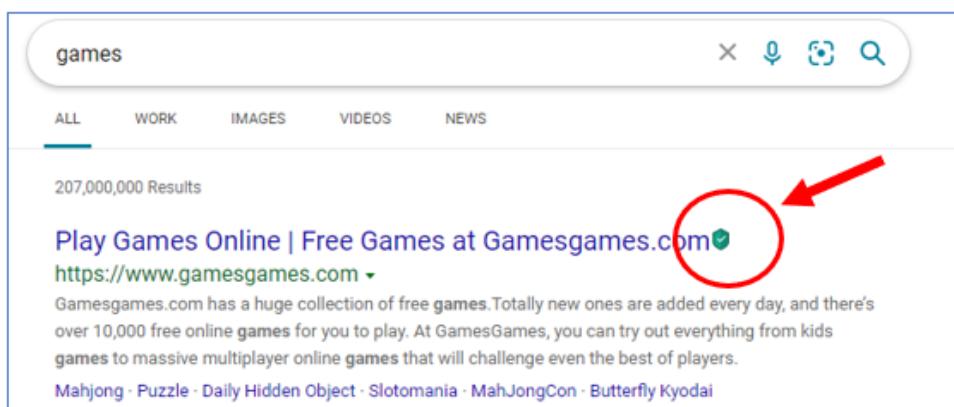


Figure 1. 4 Flagging websites that aren't secure

3) FTP Protocol

FTP is an acronym for File Transfer Protocol. FTP allows files or data to be transferred between two computers via the internet. FTP is a TCP-based network transport protocol. TCP ensures the security of FTP. This ensures that your data is sent to its target destination. For example, someone in the globe might desire to make their files available for others to download. They only need to upload their files to an FTP server. Others can then connect to the FTP server and download the files via the FTP protocol.

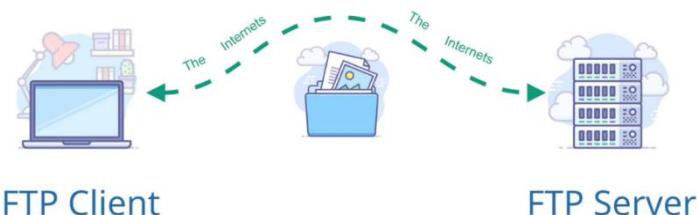


Figure 1. 5 File Transferring between FTP Server and FTP Client

Now there are couple of ways to transfer files using FTP.

(1) By using standard internet browser

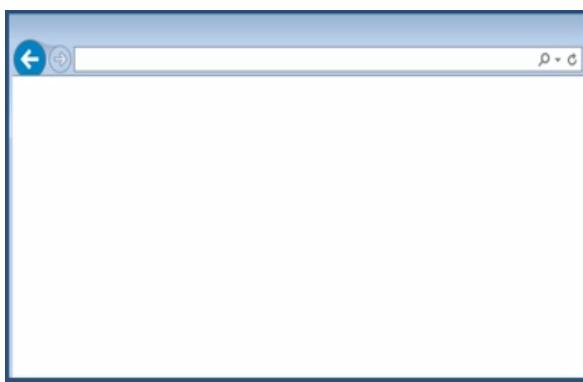


Figure 1. 7 Using standard internet browser for FTP

(2) By using FTP Client



Figure 1. 6 Using FTP Client for FTP

Using standard internet browser for FTP →

Let's download mp3 files from someone's FTP site using a typical internet browser. Just like a regular website, type the address of the FTP server we wish to connect to in the web address box. The prefix is now ftp rather than http. Now we can look through the files that the server's owner has created and download whatever we want. If desired, the owner can set up a "username" and "password."

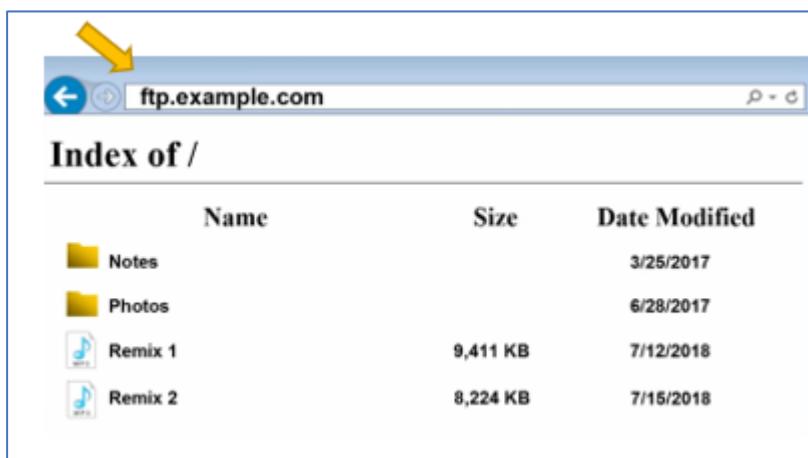


Figure 1. 8 Typing FTP address in the web address box

Using FTP Client for FTP →

There are a variety of FTP programs available, however FileZilla is the most popular free FTP client. When compared to utilizing a web browser, FTP clients offer a more graphical interface and overall experience. Many FTP clients also allow us to connect to cloud storage platforms (such as Dropbox and Google Drive), update files on a server without having to download them, and keep track of our file upload and download history.

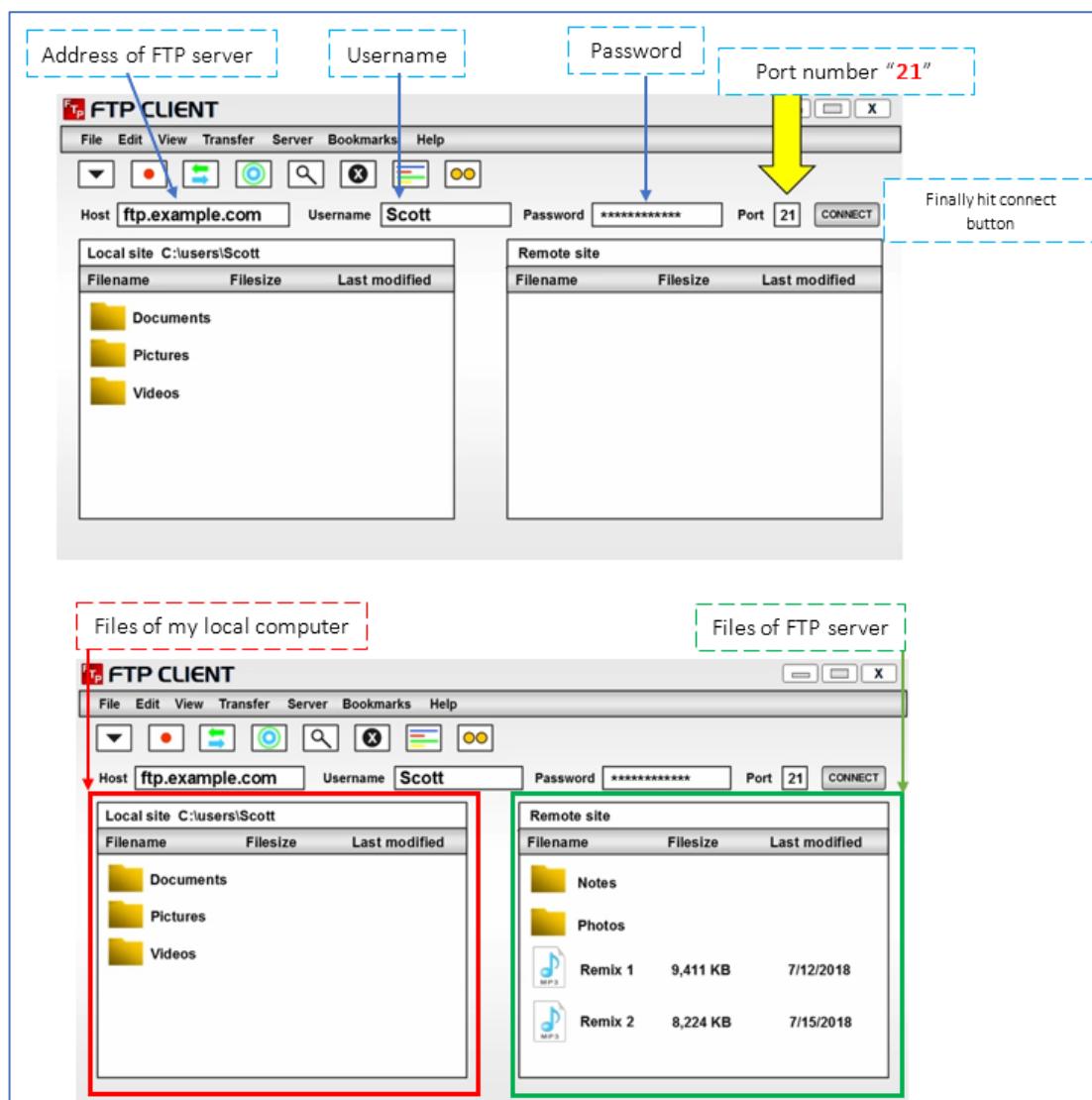


Figure 1. 9 FTP Client interface properties

We can simply drag and drop files from the FTP server to our PC files to download them. And we can simply drag and drop files from our computer to the FTP server to upload them. The most significant disadvantage of FTP is that it is not a secure protocol that employs encryption. As a result, data is transmitted in plain text.

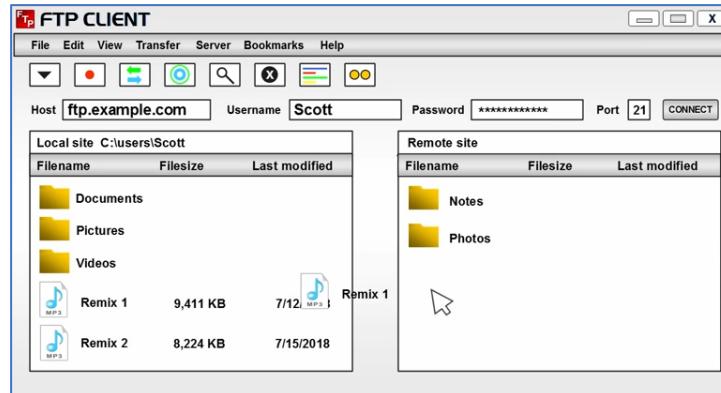


Figure 1. 10 Drag and drop files from PC to FTP Server

1.2 Web servers

1.2.1 Introduction

A server is a software component or dedicated hardware that can accept and respond to requests from numerous clients. A client is a device that sends requests to the server and receives responses. As a result, there is no such thing as a single server. There is always some kind of resource being served. A server is a centralized computer to which several clients can connect across a local area network (LAN) or the Internet. As an example, as a client, I might request a video from the YouTube server.

We can deploy a desktop computer or a mobile phone as a server since, as previously said, a server is more than a physical machine. We can use the appropriate software to put them up as servers. A desktop PC, for example, can be used as a File Server to share files with other networks.

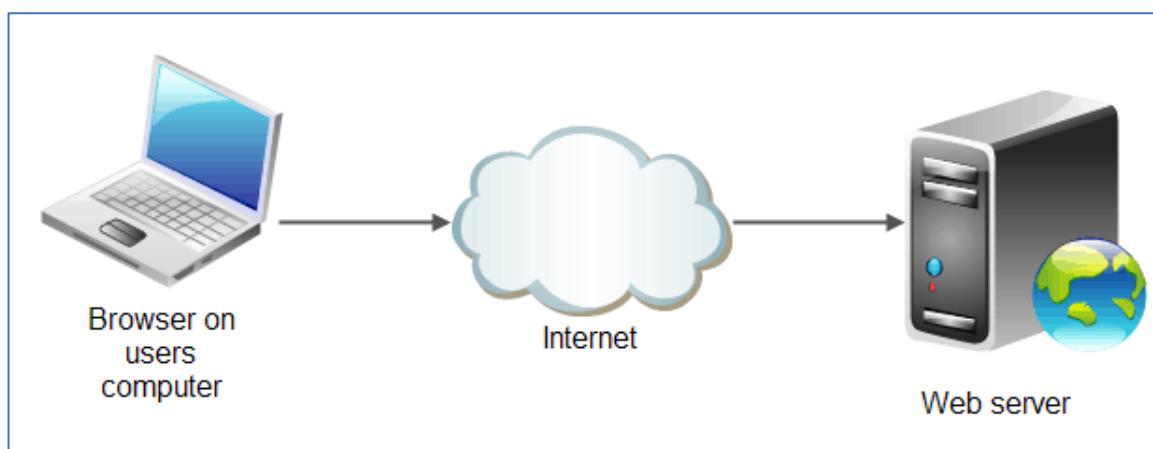


Figure 1. 11 Web Server

1.2.2 What is Web Server

A web server is nothing more than software and hardware that uses the Hypertext Transfer Protocol (HTTP) and other protocols to react to requests from clients on the World Wide Web. The webserver's major task is to show the website's content, which it accomplishes by storing, processing, and finally delivering the webpages to the user who has requested them. The web server also supports SMTP (Simple Mail Transfer Protocol), FTP (File Transfer Protocol), and HTTP (Hypertext Transfer Protocol). These are used to send and receive files, as well as to store them.

The web server hardware connects to the internet, allowing data to be exchanged with other connected devices. The web server software regulates how the user evaluates the files that have been hosted. The web server process is a good example of the client-server model. Webserver software should be installed on all machines that host websites. Web servers are used in online hosting, data hosting for websites, and many types of web-based applications.

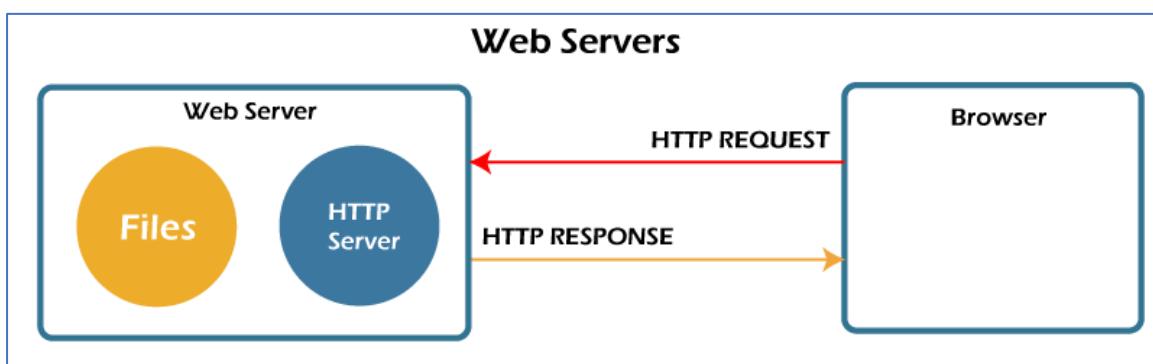


Figure 1. 12 Web Server process in summary

1.2.3 Web Server Process

A website's domain name is used to access the web server software. This ensures that the site's content is sent to the user who requested it. The webserver's software consists of a number of different components, with at least one HTTP server. The URLs and HTTP are understood by the HTTP server. The web server hardware is essentially a computer that will contain the webserver software as well as the website's files. Documents, HTML, JavaScript files, and images are among them.

When it comes to the web server procedure, If a web browser, such as Firefox or Google Chrome, requests a file from a web server, the browser uses HTTP to request the file. When the web server receives the request, the HTTP server will accept it, find the content, and return it to the browser through HTTP.

When a browser requests a page from a web server, a series of actions will occur. A person will first type a URL into the address bar of a web browser. After that, the web browser will either translate the URL through DNS (Domain Name System) or search its cache for the IP address of the domain name. The browser will be directed to a web server as a result of this action. The browser will then use an HTTP request to request the appropriate file from the web server. The web server will react by delivering the requested page to the browser through HTTP once more. The web server will respond with an error message if the requested page does not exist or if something goes wrong. The webpage will then be shown by the browser. Furthermore, the web server is capable of hosting several domains.

(Gillis, 2020)

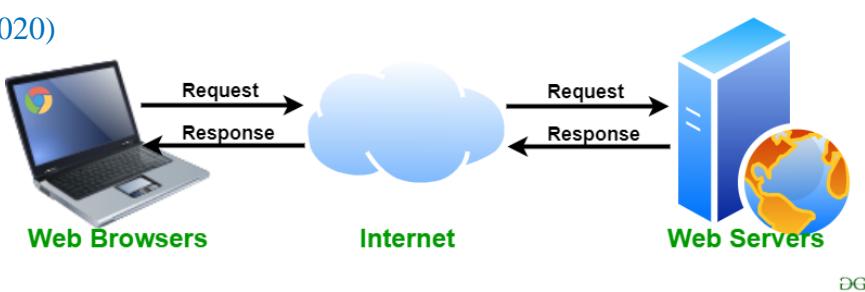


Figure 1. 13 Web Server Process

1.2.4 Dynamic vs Static web servers

Static and dynamic material can both be served by a web server. Static material is displayed as is, but dynamic content can be updated and changed. A Computer and HTTP software will make up a static web server. Because the server will provide hosted files to a browser in their current state, it is termed static.

A web server and associated applications, such as an application server and a database, will be included with dynamic web browsers. Because the application server can be used to update any hosted files before they are transmitted to a browser, it is termed dynamic. When material is requested from the database, the web server can generate it. This method is more flexible, but it is also more difficult.

Table 1. 1 Dynamic vs Static web servers

| Static Website | Dynamic Website |
|---|--|
| A website with HTML-coded web pages, where the content of each page is fixed and does not change unless modified and republished. | a website that generates web pages in real time |
| Client-side technologies such as HTML and CSS were used to create this website. | Client-side technologies such as server-side scripting languages were used in the development. |
| Unless the source code has changed, the content remains unchanged. | The content changes in response to client requests. |
| Simple and straightforward to program. | Programming is difficult and complex. |
| Doesn't allow for a lot of user engagement. | Allow for a lot of user engagement. |

| | |
|--|--|
| Host is cheaper. | Host is costly. |
| Difficult to update. | Easier to update. |
| These are used for small scale website which doesn't require continuous changes. | These are used for large scale websites which require continuous changes. |
| resume websites, portfolio websites, brochure websites, one-off landing pages, and other informational or read-only sites. | writing blogs, e-commerce sites, calendar, to-do sites and other types of sites which needs updating frequently. |



Figure 1. 14 Dynamic vs Static web sites

1.2.5 Web server examples

1) Apache HTTP server →

This is the most widely used web server, with about 60% of the world's web server computers running it. The Apache Software Foundation created the Apache HTTP web server. It is open-source software, which means we can access and modify its code to suit our needs. The Apache Web Server is simple to set up and use on practically any operating system, including Linux, MacOS, and Windows.

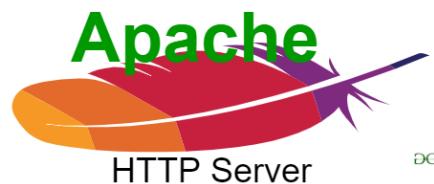


Figure 1. 15 Apache HTTP server logo

2) Microsoft Internet Information Services (IIS) →

Microsoft's IIS (Internet Information Services) is a high-performance web server. It is strongly connected to the operating system, making it relatively easier to manage. It was created by Microsoft, and it has a strong customer support system that is simple to use if we have any problems with the server. It contains all of the capabilities of the Apache HTTP Server, with the exception that it is not open-source software, and so its code is inaccessible, preventing us from making changes to fit our needs. And it is simple to install on any Windows device.



Figure 1. 16 Microsoft Internet Information Services (IIS) logo

3) Lighttpd →

The name Lighttpd is pronounced 'Lightly.' It presently hosts roughly 0.1 percent of all websites on the internet. Lighttpd uses a little amount of CPU resources, making it easy to run. It has a small memory footprint and hence requires less memory space to execute than other web servers, which is always a benefit. It also features speed optimizations, which means we may adjust or vary the speed based on our needs. It's open-source software, which means we may look at the code and make changes to suit our needs before uploading our own module (the changed code).



Figure 1. 17 Lighttpd Server logo

4) Jigsaw Server →

Jigsaw is a Java-based application that can run both CGI (common gateway interference) scripts and PHP programs. It was created as an experimental server to demonstrate new web protocols and is not a full-fledged server. It's open-source software, which means we may look at the code and make changes to suit our needs before uploading our own module (the changed code). It can be installed on any device that supports the Java programming language and Java customizations.



Figure 1. 18 Jigsaw Server logo

5) The Sun Java System →

Python, PHP, and other Web 2.0 languages, scripts, and technologies are supported by the Sun Java System. It isn't open-source software, thus its code isn't available, which means we can't update it to fit our needs.



Figure 1. 19 The Sun Java System logo

1.3 Servers

1.3.1 Introduction

Server could be a software or hardware that accept requests from multiple clients to provide immediate responses. In simple terms, Server is a device or a software which provide services to client. This is a centralized machine where multiple clients can connect through LAN or from the internet.

Client is a device which make requests or client is a device which accept responses from a server. For example, students as a client request a video from YouTube server. We can set up even our own Desktop PC or mobile phone as a server. Because as told before, server is not just a physical computer. We can set up them as servers with right software's. For example, Desktop PC can be used as a File Server to share files to other networks.

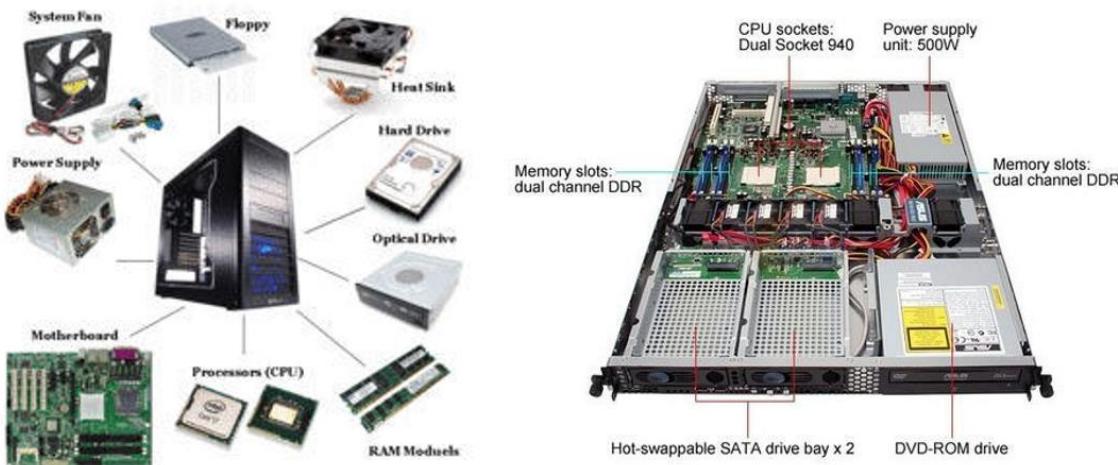


Figure 1. 20 Parts of a Server

1.3.2 Main 3 Types of Servers

Some servers are committed to do specific tasks. For example, some servers can be dedicated to website, some can be dedicated to store files likewise. Big organizations tend to have multiple servers to handle these tasks. But in small organizations they set up 1 server to handle all types of work.

There are mainly 3 types of servers.

1. Physical Servers (Dedicated Server) – Dedicated servers which use traditional CPU and RAM components. This is a type of server which buyer purchase hardware and store inside home or business area. Hence the buyer has physical access to its resources.
2. Virtual Server – This is a part of physical server. Virtual server is representing physical server and buyer doesn't have to purchase hardware. Most servers which are available online are Virtual Servers. This is a web server which shares computer resources such as CPU, RAM and storage space with operating system (OS). Hence the buyer doesn't have physical access to its resources.
3. Hybrid Server – Hybrid server is more like a dedicated server, but vendor provides a slice of dedicated server. But it has flexibility of cloud, but with the power of dedicated. Buyer have to share the hardware with other customers of the vendor. Buyer have to share the hardware (CPU, RAM's, processor) but hard drives aren't shared with others. Hence the buyer's data will always be safe and secured. The data is kept separately, and everybody got their own credentials.

There are types of servers which can perform different kind of functions. Below down here are the common server types which are specialized for different functions.

1Application Servers

Application Server handles all application operations between end users and an organization's backend business applications data. They occupy large computing territory between database servers and the end user. There are many Application Server platforms such as JBoss, WebLogic, J2EE, Glassfish, Tomcat.

2.FTP Server

FTP (File Transfer Protocol) Servers are built to handle data transfer between computers. These servers allow users to browse, upload and download files as they wish. This can be done by authentication through FTP client. Some of the FTP clients are FileZilla Client, FTP Voyager and Core FTP.

3.File Servers

This is different from FTP Server. This is a common server being used by lot of companies. This is specifically made for storing files and folders. Hence, companies tend to save business data and documents on file servers. Commonly File Servers use protocols such as SMB (Server Message Protocol), NFS (Network File System).

4.Print Servers

These servers help to manage and distribute printing functionality instead of attaching printers for every workstation. Hence, a single print server can accept multiple printing requests from multiple clients at the same time.

5.DNS Servers

Larger business groups tend to have DNS servers. DNS (Domain Name System) convert host names to machine readable IP addresses. These servers contain database of public IP addresses with their associated host names. Hence, they help to resolve these names into IP addresses when requested.

6. DHCP Servers

DHCP (Dynamic Host Communication Protocol) server assigns IP addresses and default gateways automatically. This saves from troubles of manually configuring static IP addresses and other network settings to computers.

7.Mail Servers

These servers are used send and receive emails. These servers use email protocols such as SMTP(Simple Mail Transfer Protocol), IMAP (Internet Message Access Protocol) and POP3(Post Office Protocol version 3).

8.Web Servers

These servers are commonly used in today's market. This type of servers is specialized for storing, processing, and delivering web pages to users. This mainly uses HTTP protocol (Hypertext Transfer Protocol) for this intercommunication. Some of common Web Servers are Apache, Nginx, and Microsoft Internet Information Services.

9.Database Servers

Database Servers used to store and manage databases. These servers provide data access to authorized users, and they keep data in centralized location for back up. These servers are specially designed to respond multiple requests from many clients at the same time. Some of the common Database Servers applications are Oracle, MYSQL and Microsoft MQL.

10.Proxy Servers

Proxy Servers can translate traffic between networks and protocols. Since they provide gateway between users and internet, Proxy Servers helps to prevent cyber-attacks. Hence, they can act as a firewall and web filter. And also, Proxy Servers can cache data to speed up requests.

11.Real-Time Communication Servers

Real-time communication servers, also known as chat servers or IRC servers, and still known as instant message (IM) servers, allow large groups of people to communicate almost instantly.

12.Telnet Servers

A Telnet server allows users to connect to a host computer and carry out functions as if they were on the remote computer.

13.Virtual Servers

In 2009, the number of virtual servers deployed outnumbered physical servers for the first time. In today's data center, server virtualization is nearly widespread. Server Watch examines the newest virtualization technology trends, from hypervisors to hybrid clouds.

1.3.3 Other Types of Servers

1. Exchange Server :

Microsoft Exchange provides email, calendar, and contacts on our PC, phone, and web browser, and is designed to provide the enterprise-grade security and reliability that organizations demand.

2. Lync Server :

Lync Server is a real-time communications server that enables enterprise instant messaging, presence, VoIP, ad hoc and scheduled conferences (audio, video, and web conferencing), and PSTN access via a third-party gateway or SIP trunk.

3. SharePoint Server :

A SharePoint Foundation-based server product that provides a uniform, recognizable foundation for lists and libraries, site administration, and site customisation. SharePoint Server contains all of the features and capabilities of SharePoint Foundation, as well as Enterprise Content Management, business analytics, business searches, private sites, and Newsfeed.

4. SQL Server :

SQL Server is a relational database management system (RDBMS) developed by Microsoft for use in the enterprise. As a database, it is a software application whose main purpose is to store and retrieve data as required by other software applications, either on the same computer or on another computer across a network (including the cloud).

5. Windows Server :

Windows Server 2012, the server version of Windows 8, redefines the server category with hundreds of new capabilities and advancements covering virtualization, networking, storage, UX, cloud services, automation, and more.

1.3.4 Server Clusters

As the name implies, a cluster is made up of two or more computers that are all working at the same time. Each computer in a cluster is referred to as a node, and it has its own set of characteristics such as hard disks, RAM, and CPUs. Despite the fact that each server cluster has its own resources, it is identified in the network by a single host name and a single IP (Internet Protocol) address. A server cluster is a collection of servers that work together to provide clients with high-quality services. Working in a server cluster has the advantage of allowing work to be transferred to another machine in the cluster if one in the cluster fails. The first is active, whereas the second is passive. When the active server is unavailable, the passive server is activated). Clients get 24/7 access to all required server resources thanks to the server clusters.

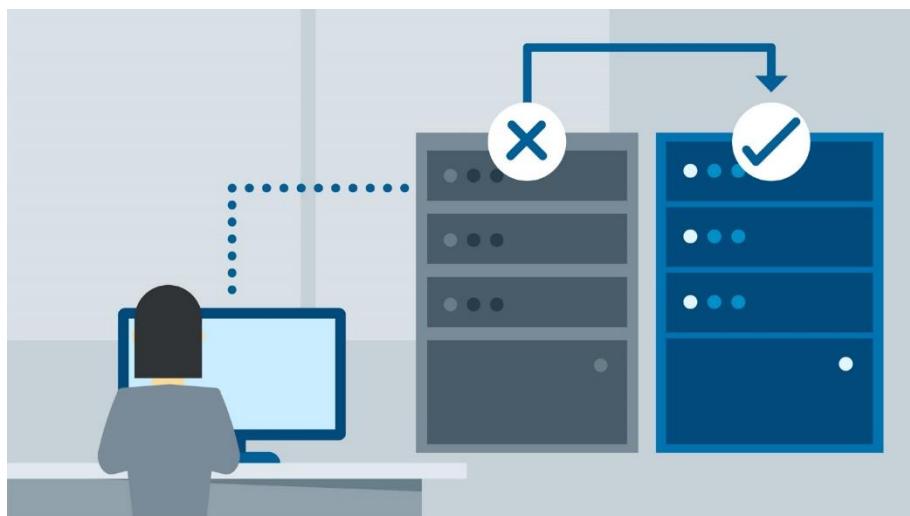


Figure 1. 21 Server Clusters

1.4 Server Hardware Types

1.4.1 Introduction

When decide to buy a new server, there are a lot of technical specifications that have to deal with. How a server looks from the outside is called “Form Factor.” With known in the form factor helps to estimate the size and the shape of a server.

There are several Forma Factors of a server.

1. Tower Server
2. Rach Server
3. Blade Server

Tower Servers :

General standing up servers called as Tower Servers. They have built in as an upright cabinet which very similar to Desktop PC but slightly larger. Because of the independent design of Tower Servers, they can add to networks easily. When compared to other servers, Tower Servers require less maintenance.



Figure 1. 22 Tower Server

Rack Servers :

They are designed to be kept inside racks. Though they can be kept wherever wanted such as slide under TV stand. Since these servers are designed to keep in racks, the size of these servers is specified in terms of Rack Unit (RU).

Mainly the cooling fans generate a noise in Rack Servers. They are noisier than Tower Servers. Since rack server components are packed more densely than the Tower Server, the cooling fans of the Rack Server have to do more work when it comes to cool the components.



Figure 1. 23 Rack Server and how Rack Server mount in a rack

Blade Servers :

Blade servers are designed to minimize the use of physical space hence they have stripped down architecture. With a blade server it is mandatory to buy blade enclosure with it. Which means Blade Servers can only run with compatible Blade Enclosure. Because Blade Enclosure provide features of power, cooling, and networking all the Blade Servers are packed in.



Figure 1. 24 Blade Server and how Blade Server mount in an enclosure

1.4.2 How to understand Server Hardware Name and Brands

The name of the server tells few things about CPU, supported hardware and expansion capabilities of the Server. Let's consider 2 famous brands of servers which are HP ProLiant and Dell Power Edge.

The 1st example is **HP ProLiant**.

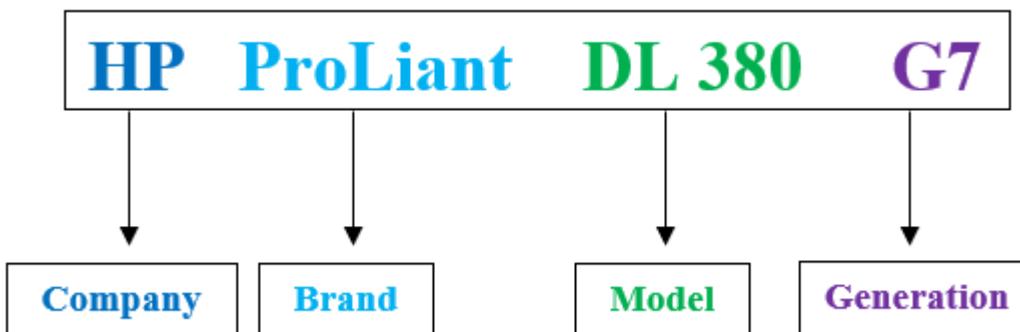


Figure 1. 25 Understand Server Name and Brands of HP ProLiant

Since Company and Brand self-explanatory, let's see Model. In model number DL stands for Density Line, which is another name for rack mountable server. Below figure shows other product lines of ProLiant servers. All these indicate the form factor of the server.

Table 1. 2 HP ProLiant product lines with their form factor

| Product Line | Meaning | Form Factor |
|---------------|---------------|---------------------------------|
| ML | Modular Line | Tower. |
| DL | Density Line | Rack based for general purpose. |
| SL | Scalable Line | Rack based for data centers. |
| BL | Blade Line | Enclosure based. |
| Micro Servers | Entry Level | Small form factor. |

In model number 380 can break down into 3 constituent numbers as below.

$$380 \rightarrow 300 + 80 + 0$$

300 is the Series. This indicates the number of processor sockets supported by the server. In HP ProLiant 300 series supports up to 2 processor sockets. That means it can have 2 physical CPUs in that server. Below figure shows other series of ProLiant servers with the number of supporting processor sockets.

Table 1. 3 HP ProLiant different series with number of CPU sockets

| Series | Number of CPU sockets |
|--------|-----------------------|
| 300 | 2 |
| 500 | 4 |
| 900 | 8 |

80 is a variation of 300 series. In 300 series there are 4 variants. They are 320, 360, 370 and 380. They each are different in terms of hardware as below picture shows.



Side-by-Side Comparison

| | HP ProLiant DL360 Gen 9 | HP ProLiant DL380 Gen 9 |
|-------------------------|----------------------------------|----------------------------------|
| Processor | Intel Xeon | Intel Xeon |
| Maximum memory | 3.0 TB | 3.0 TB |
| Form factor | 1U | 2U |
| LFF HDDs | 4 | 12 |
| SFF HDDs | 10 | 24 |
| Cores per Socket | 22 | 22 |
| Memory | DDR4 | DDR4 |
| Upgradeability | Up to 2 processors (44 cores) | Up to 2 processors (36 cores) |

Figure 1. 26 Difference between server variants

And the last digit which is 0 in this example indicate the processor type. 0 means it uses Intel processors. If the last digit is 5, it uses AMD processors.

The last part of the server's name is the Generation. The giving example is G& which means Generation 7. The G1 which means 1st Generation of HP ProLiant DL380 came in 1990s which had only 3 Pentium processors and supported up to 4GB of RAM. And the latest generation of HP ProLiant is 10th Generation which is indicate as G10. It can support up to 4TB of RAM and having Xeon processor which can go up to 28 cores as below table indicates.

Table 1. 4 DL 380 difference between generation 1 and generation 10

| DL 380 | GEN 1 | GEN 10 |
|--------------------------|-------------------|------------|
| Processor Support | Intel Pentium III | Intel Xeon |
| Cores supported | 2 | 28 |
| Maximum RAM capacity | 4GB | 4TB |
| Maximum internal storage | 436.8GB | 459GB |

The 2nd example is **Dell Power Edge**.

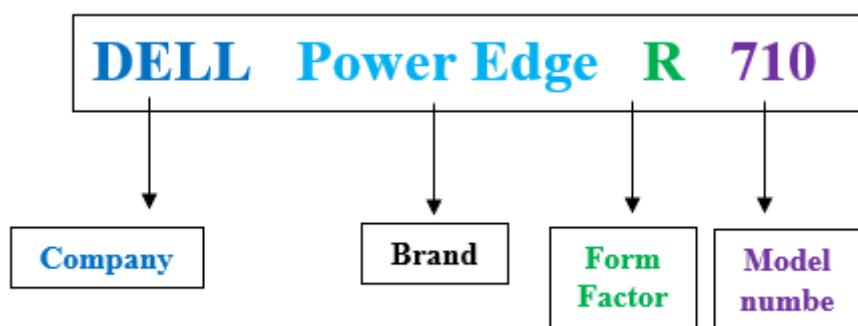


Figure 1. 27 Understand Server Name and Brands of Dell Power Edge

As HP ProLiant example the Company and Brand self-explanatory, let's see Form Factor. As in the above example server name the “R” indicates Rack Mountable server. And “T” indicates Tower server.

In the model number part, 1st digit indicates the number of CPU sockets. As below table show number of CPU sockets can be determine the value of the 1st digit. In our example the 1st digit is 7 which means according to below table it has 2 CPU sockets.

Table 1. 5 Dell PowerEdge First Digit vs the number of CPU sockets

| Value of first digit | Number of CPU sockets |
|----------------------|-----------------------|
| 1-3 | 1 |
| 4-7 | 2 |
| 8 | 2 or 4 |
| 9 | 4 |

The 2nd digit indicates the Generation. 0 means it is 10th generation. And as in our example 1 means 11th generation.

And the last digit which is 0 in this example indicate the processor manufacturer. 0 means this server uses Intel processors. If the last digit is 5, it uses AMD processors.

1.4.3 Things to know before buy a Server Hardware

When it comes to buy a server there are 4 options to choose as following.

1. New
2. Refurbished
3. Used
4. Buy for parts

Buying a New Server → New servers are the costliest among them. But most of time with new servers can get 3-5 years support contract with the vendor. This will ensure the quick availability of any failed components and on-site support if run into any issues. HP provide firmware updates only if buyer has support contract with them.

Buying a Refurbished Server → Refurbished servers are the servers which have previously returned to the manufacturer for various reasons but not sold in the market. During refurbishing the seller will ensure the server is ready to deploy by cleaning the server, testing the components, ensuring the firmware updated and leave everything at its factory defaults. They are cost effective when compared to new servers.

Buying a Used Server → Used servers usually don't have refurbishment. And they have no warranty. Used servers are even cheaper than refurbished servers. But it is risky to buying it since there are no warranty.

Buying Server parts → When people want to build their own server or when want to upgrade their server, they are looking for buying server parts.

1.4.4 Other things to consider when buying a Server Hardware

Budget :

Budget is important since it helps to control the spending and to focus on long term financial investment. Since servers are too expensive, it's necessary to track expenses of them. There are many servers with a range of price depending on specifications. If the running business is a long term, a suitable server will help to improve efficiency of the business. Considering the amount of spending money for the server is very important. Buyer can get very expensive server or cheaper server as prefer.

Table 1. 6 Pros and Cons of buying cheaper server and expensive server

| Buying a cheaper server | Buying an expensive server |
|---|--|
| When renting cloud server may under \$100 for month. | When renting cloud server may over \$100 for month. |
| When buying dedicated may under \$500 for 1 time payment. | When buying dedicated may over \$1500 - \$2000 for 1 time payment. |

Small servers can buy for few hundred dollars but more powerful servers such as cloud servers can cost that much for per month or even thousands for on-site servers. Because of that. Buyer should buy a server for affordable price. It would be a waste of money for spending excessive amount of money for a server that exceed buyer requirements.

Server's main function :

Server could be different kind of computer or a software that specialized for various range of tasks. Server components such as RAM, processor, storage capacity play a major role before buying a server. Buyer has to consider details about these components before buy a server. Since different kind of servers performing different kind of functions, it is mandatory to buy a server which is matching buyer's requirement. Otherwise, it would be a waste of huge amount of money.

Server Type :

There are 2 types of servers as mentioned before. Physical Servers (On-Site Servers) and Virtual Servers (Cloud Servers). Physical servers have their own advantages and disadvantages. Since physical servers require space, installation and maintenance Cloud servers seems to be convenient compared to On-Site servers. Because we can access Cloud server from anywhere. And they provide better security. But in contrast the file transfer speed would be lower. Since buyer have to consider these advantages and disadvantages before making decision to buy a server.

Table 1. 7 Advantages and disadvantages of having on-site server

| Advantages | Disadvantages |
|--|---|
| Have to pay 1 time to buy the hardware. (Enterprise grade hardware cost around \$2000 - \$3000) | To cool the server are power the server cost high electricity. (This could be cost \$200-\$400 per month) |
| No need to pay monthly fee to vendor | Need to set backup generator (This could be cost \$2500 - \$4500) |
| | Hardware replacement parts are expensive. |
| | Can't replace hardware immediately if a failure happens. |
| | Not having 24 x 7 customer support. |
| | Hardware becomes outdated quickly. |

Table 1. 8 Advantages and disadvantages of having cloud server

| Advantages | Disadvantages |
|---|--------------------------------------|
| Can have latest hardware | No need to pay monthly fee to vendor |
| Can upgrade and customize the hardware as buyer's requirement | Can't access server physically. |
| Having 24 x 7 customer support. | |
| Benefit of having on-site staff for hardware replacement and maintenance. | |
| No additional charge to cool the server and power the server | |
| No need to buy backup generator | |

Server Build :

There are many preconfigured servers available in market ready to plug and play. And also, there are custom server which allows buyers to choose components and parts as they prefer. But its need expertise knowledge to decide the custom build and it would be requiring more work and effort to buy a server rather than buying preconfigured server.

Security :

Buyers has to consider the security of the server since it is dangerous to store all business data in one place and leave it open to attack. With a cloud server, buyer can customize the level of the protection by asking the provider. With the help of cybersecurity consultant, buyer can decide best suitable secured level server to operate.

Maintenance :

Physical Servers need to be monitored and require maintenance which costing lot of money to do that. Hence, business should hire onsite IT support to handle Physical Servers. But with having a Cloud Server, buyer doesn't need to maintain and handle them since the service provider handles it.

1.4.5 Server Hardware Prices

Tower Server Prices

HPE ProLiant ML350 Gen10

- Condition : Refurbished
- Processor 2x 6254 Xeon Gold (3.1 GHz / 24.75 MB Cache)
- Hard Drive 8x 1TB SATA 3.5" Hard Drives
- Memory 128GB (8x16Gb) Memory Installed - DDR4
- Power Dual 500W Power Supply
- Other : Battery, Fans, Heatsink included
- Price : \$11 274

(Ebay, n.d.)



Figure 1. 28 HPE ProLiant ML350 Gen10

Dell PowerEdge T330

- Condition : Used
- Processor E3-1230 v5 with 4 Cores 3.40GHz
- Hard Drive 8x 1TB 7.2k 6Gbs SATA
- Memory 32GB Memory Installed - DDR4 U
- Power Supply 2x 495W
- Other : DVD drive, Heatsink included
- Price : \$849

(Ebay, n.d.)



Figure 1. 29 Dell PowerEdge T330

Lenovo Think System ST50

- Processor E-2124G Intel Xeon (3.4GHz 8MB Smart Cache)
- Memory : 8GB Memory Installed - DDR4-SD
- Hard Drive : Not installed
- Power Supply : 250W
- Other : DVD drive
- Price : \$1 008

(Ebay, n.d.)



Figure 1. 30 Lenovo Think System ST50

Rack Server prices:

DELL PowerEdge R440 Server

- Condition : Refurbished
- Processor : 2x 4108 Xeon Silver (1.8 GHz / 11MB Cache)
- Hard Drive : 4x 1.92TB SATA 2.5" Hard Drives
- Memory : 256GB Memory Installed - DDR4
- Power : Dual 550W Power Supply
- Other : Battery, Fans, Heatsink included
- Price : \$3 607



Figure 1. 31 DELL PowerEdge R440 Server

(Ebay, n.d.)

HP ProLiant DL380p G8 Server

- Condition : Refurbished
- Processor : Intel Xeon e5-2620v 6 Core (2.1Ghz/15Mb Cache)
- Hard Drive : 25x 146GB 6G SAS
- Memory : 192GB Memory Installed - DDR3
- Power : Dual 750W Power Supply
- Other : Battery, Fans included
- Price : \$4 014



Figure 1. 32 HP ProLiant DL380p G8 Server

(Ebay, n.d.)

HP ProLiant DL380 Gen10

- Condition : Refurbished
- Processor : 2x 5218R Gold (2.1 GHz 20C 27.5MB Cache)
- Hard Drive : 24x 240GB SATA 2.5" Hard Drives
- Memory : 384GB Memory Installed - DDR4
- Power : Dual 800W Power Supply
- Other : Battery, Fans, Heatsink included
- Price : \$8 323



Figure 1. 33 HP ProLiant DL380 Gen10

(Ebay, n.d.)

Blade Server Prices

Dell PowerEdge M520

- Condition : Refurbished
- Processor : 2 x Intel Xeon Quad-Core E5-2407 (2.2GHz)
- Hard Drive : 2 x Dell 300GB 15K SAS
- Memory : 16GB DDR3 RAM (2x 8GB)
- Price : \$526

(Ebay, n.d.)



Figure 1. 34 Dell PowerEdge M520

HP ProLiant BL460c Gen8

- Condition : Used
- Processor : 2 x Intel Xeon Quad-Core E5-260 (2.4GHz)
- Memory : 16GB DDR3 RAM (2x 8GB)
- Price : \$477

(Ebay, n.d.)



Figure 1. 35 HP ProLiant BL460c Gen8

Fujitsu PY Primergy BX924 S4

- Condition : Used
- Processor : 2 x Intel E5-2667v2 8-Core XEON (3.3GHz)
- Memory : 192GB DDR3 1866MHz Fujitsu original memory
- Price : \$3 289

(Ebay, n.d.)



Figure 1. 36 Fujitsu PY Primergy BX924 S4

Cisco B200 M3

- Condition : Used
- Processor : 2 x Intel Xeon 8-CORE E5-2650 (2.20GHz)
- Hard Drive : Not installed
- Memory : 96Gb DDR3 RAM
- Price : \$1 644

(Ebay, n.d.)



Figure 1. 37 Cisco B200 M3

Cloud Server Prices

Table 1. 9 Cloud Servers price by “Server Mania”

| Cores | RAM | Storage | Bandwidth | Monthly Rent |
|--------|-------|---------|-----------|--------------|
| 1 CPU | 1GB | 25GB | 1TB | \$5 |
| 1 CPU | 2GB | 50GB | 2TB | \$10 |
| 2 CPU | 4GB | 80GB | 4TB | \$20 |
| 2 CPU | 6GB | 100GB | 4TB | \$30 |
| 4 CPU | 8GB | 160GB | 5TB | \$40 |
| 6 CPU | 16GB | 320GB | 8TB | \$80 |
| 8 CPU | 32GB | 640GB | 12TB | \$150 |
| 12 CPU | 48GB | 960GB | 16TB | \$250 |
| 16 CPU | 64GB | 1.5TB | 20TB | \$350 |
| 20 CPU | 96GB | 2TB | 20TB | \$500 |
| 24 CPU | 128GB | 3TB | 20TB | \$650 |

(Server Mania, n.d.)

Hybrid Server Pricing

Table 1. 10 Hybrid Servers price by “Server Mania”

| Series | V Cores | RAM | Storage | Cloud Backup | Monthly Rent |
|--------------|---------|------|-----------|--------------|--------------|
| E3 VM SSD L1 | 2 | 8GB | 120GB SSD | 50GB | \$30 |
| E3 VM SSD L2 | 2 | 8GB | 250GB SSD | 50GB | \$40 |
| E3 VM SSD L3 | 2 | 8GB | 500GB SSD | 50GB | \$50 |
| E3 VM SSD P1 | 4 | 16GB | 250GB SSD | 50GB | \$60 |
| E3 VM SSD P2 | 4 | 16GB | 500GB SSD | 50GB | \$70 |
| E3 VM SSD P3 | 4 | 16GB | 1TB SSD | 50GB | \$80 |

(Server Mania, n.d.)

1.5 Server Operating System

1.5.1 Introduction

This is a type of operating system that is meant to be used on servers. It's used to provide services to a large number of people. It's a cutting-edge operating system that can handle several customers at once. It's a more powerful operating system with features and capabilities that are necessary in a client-server architecture or similar enterprise computing environment.

A server is a computer that allows other devices to access data. It can send data over the Internet to LAN or WAN systems. A Web server, for example, can run Apache HTTP Server or Microsoft IIS, both of which provide access to websites on the Internet. Windows, file, and database servers, Web server, mail server, application server, and print server are all supported and enabled by server operating systems. It is a more complex version of the fundamental operating system that handles network requests from server-client machines. It comes with everything we need to work in a client-server computer environment.



Figure 1. 38 Server Operating System types

1.5.2 Server Operating System Features

The server operating system has a number of features. This runs all or most of the processes specified by the operating system command. A server operating system is meant to execute huge programs and carry out complex tasks such as data transfers. It has GUI (Graphical User Interface) and command-level access to the server. And all of the administrative tools required to run a wide range of apps and duties are included in the server operating system.

The Server Operating System provides a centralized interface for managing users, security, and other administrative tasks. These operating systems assist in the construction of a stable environment in which all applications and processes can function without interruption. An operating system that is still in development or has faults and malfunctions might cause procedures to slow down and data to be lost. It allows you to configure complex hardware, software, and network settings.

Not only that, but these operating systems also administers and monitors the machines and operating systems of the client. Because servers are vulnerable to infiltration and infection, a server operating system with a built-in firewall and security features is important. When a solid security system is in place, hackers and malicious apps will have a difficult time infiltrating the client's operating system.

1.5.3 Server Operating System Benefits

A server operating system has several advantages. The main advantage is increased efficiency. It contributes to less reliance on physical servers, resulting in larger savings. The second advantage of Futureproofing is that it provides long-term sustainability by assessing the anticipated growth of the corporate network. The following advantage is increased savings due to lower downtime. Any firm that can reduce downtime saves money. The final advantage is improved performance. A faster network-supporting system will support more clients and storage solutions than another.

1.5.4 Server Operating System Types

Server operating systems come in a variety of types. The following are a few of them:

1) The Microsoft Windows Server Operating System

The Microsoft Windows Operating System is a group of operating systems created by the company Microsoft. It lets users to play games, watch videos, listen to music, save files, and execute software. It was created for use in both personal and professional settings. Other series in the Windows OS family include Windows 2000, Windows 2003, and more. Virtual memory management, a graphical user interface, multitasking, and compatibility for a variety of external devices are all included. Windows operating systems are available for desktops and laptops, servers, and mobile devices.

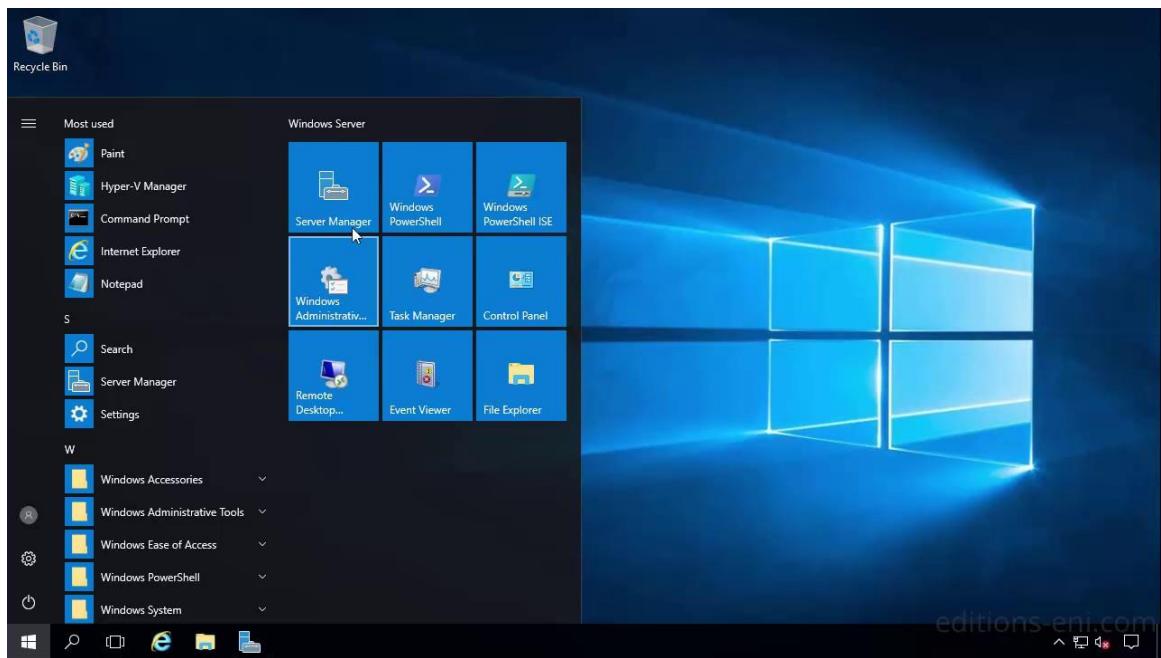


Figure 1. 39 Windows Server 2016 interface

2) Linux Server Operating System

Linux is a free operating system that performs all of the functions of UNIX. In 1991, students from Finnish universities released version 0.11, which was originally developed by them. Linux is one of the best operating systems available. It's open source, supports multi-user, multi-process, multi-thread, has good real-time performances, is powerful and robust, and is free to download from the CNU Free Software Foundation under the GPL. Application applications, such as a text editor and a high-level language compiler, are included in the OS software package, allowing the system to be controlled through windows, icons, and menus.

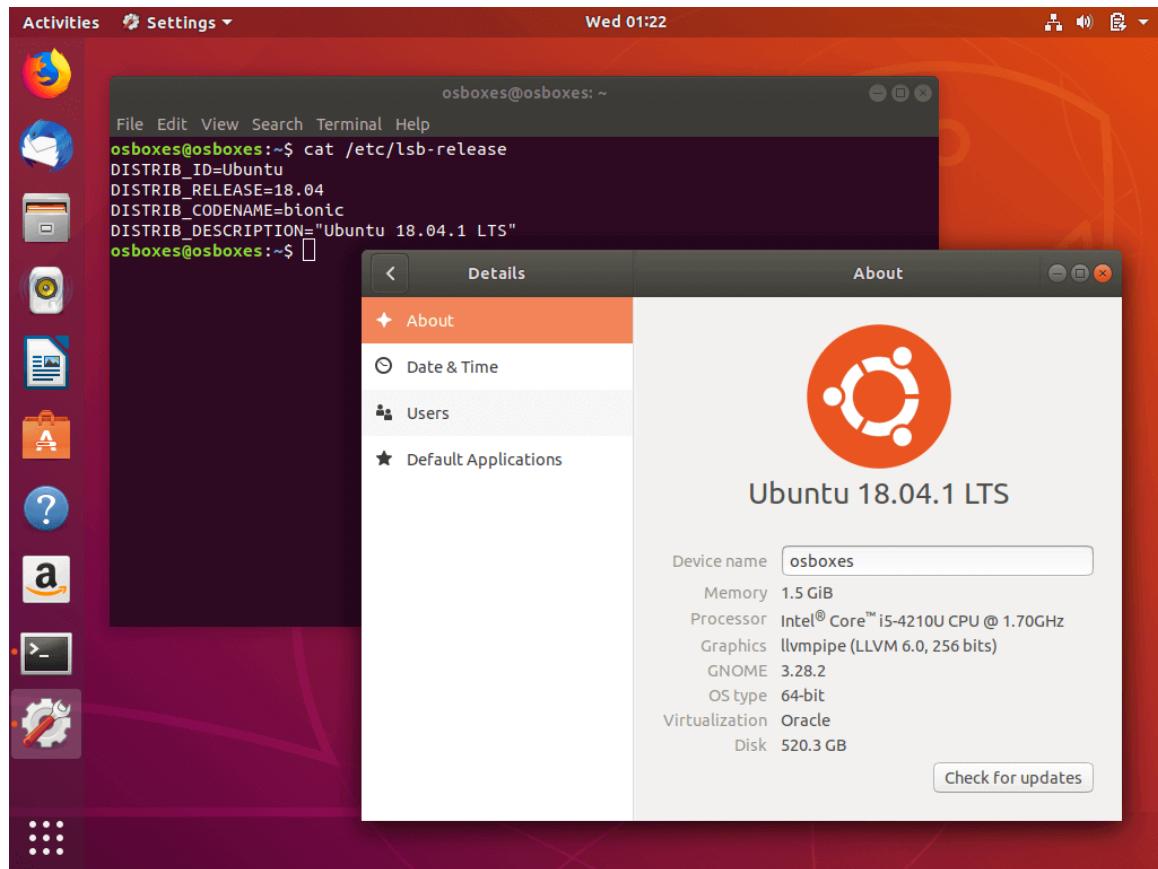


Figure 1. 40 Linux Server Operating System interface

3) UNIX Operating System

UNIX started as a time-sharing operating system for smaller computers, but it has now evolved into one of the most popular client-server operating systems. It's written in C, and the C language abbreviator functions on a variety of platforms. Other operating systems have been deployed to a smaller number of machines than UNIX. UNIX OS is a multi-user operating system with built-in TCP/IP support, as well as a high level of reliability and security. More than 90% of the countless sites that would provide services via the Internet currently use the UNIX operating system. UNIX, unlike other operating systems, is sold by a variety of companies, and there is no such thing as a true UNIX.

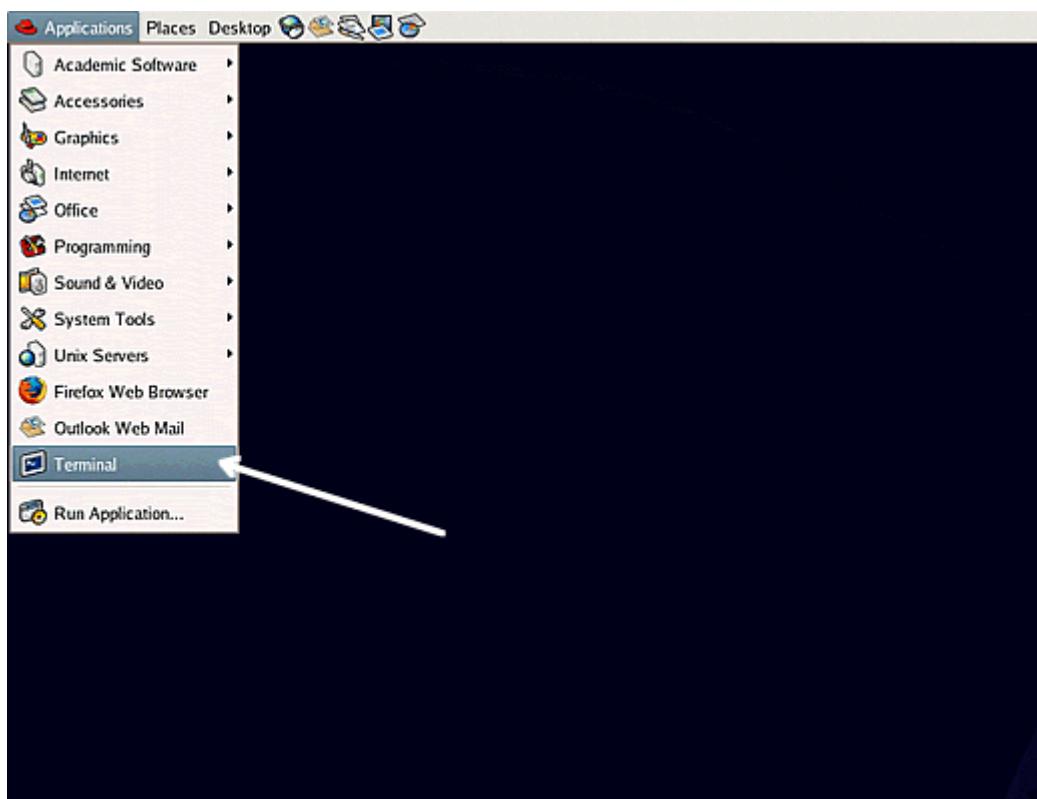


Figure 1. 41 UNIX Operating System interface

4) Netware Operating System

This is a network operating system that runs on a dedicated server. The NetWare operating system was widely used in early computer networks in the local area network (LAN). It was created by Novell in the early 1980s.

Multiprocessor and large-capacity physical memory management may be supported by NetWare series operating systems. It may include shared file access and printing capabilities, as well as support for corporate network scalability via open standards and file protocols. Versions 2 and 3 of NetWare were largely used for LAN-based file and print servers which used the Internetwork Packet Exchange (IPX) protocols and were controlled using menu-driven MS-DOS programs such as fconsole, pconsole, syscon, filer, and monitor. Through Novell Directory Services, NetWare 4 delivers a more scalable approach. The hierarchical NDS allows users to discover and use shared resources independent of their real network location. NetWare 5 is the most recent release of the NetWare operating system. It includes a new multiprocessing kernel with virtual memory support, as well as native Internet Protocol (IP), Dynamic Host Configuration Protocol (DHCP), and Domain Name System (DNS).

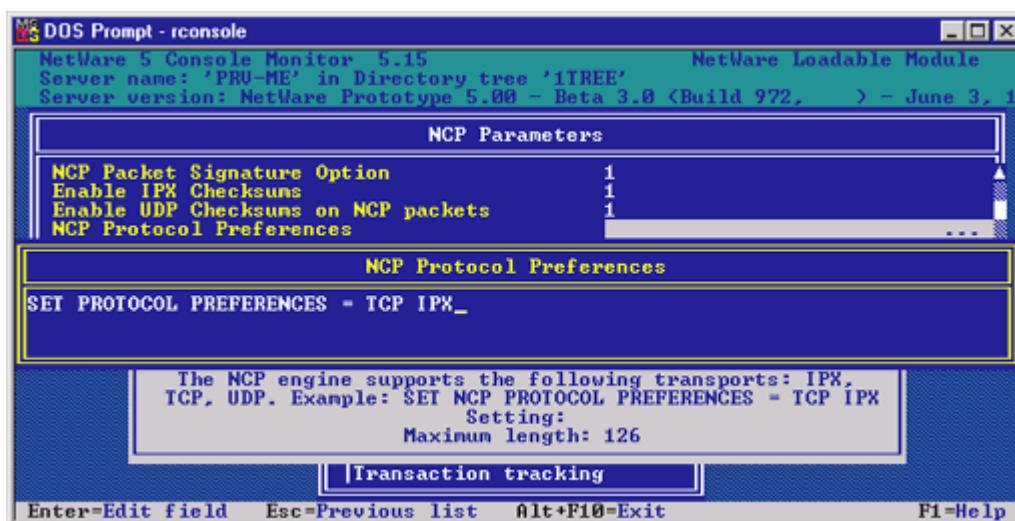


Figure 1. 42 Netware Operating System interface

5) Red Hat Enterprise Linux (RHEL)

Red Hat built this commercial Linux distribution. Red Hat Enterprise Linux comes in server and desktop variants for x86-64, Power ISA, ARM64, and IBM Z processors. The Red Hat Enterprise Linux platform serves as the hub for all official Red Hat support and training, as well as the Red Hat Certification Program. The first version of Red Hat Enterprise Linux was called "Red Hat Linux Advanced Server." In 2003, Red Hat rebranded Red Hat Linux Advanced Server "Red Hat Enterprise Linux AS," and added two new variants, Red Hat Enterprise Linux ES and Red Hat Enterprise Linux WS.

While the source code is publicly available, Red Hat maintains tight trademark laws to limit the free re-distribution of its officially approved Red Hat Enterprise Linux versions. By deleting non-free components like as Red Hat's trademarks, third-party derivatives can be developed and distributed. Scientific Linux, for example, is a community-supported distribution, whereas Oracle Linux is a commercial derivative.



Figure 1. 43 Red Hat Enterprise Linux (RHEL) interface

1.6 Windows Server

1.6.1 Introduction of Windows Server



Figure 1. 44 Wallpaper of Windows Server 2016

In basically, Windows Server is a set of operating systems developed by Microsoft primarily for usage on servers. Servers are incredibly powerful machines that are designed to function on a continuous basis and supply resources to other computers. As a result, Windows Server is virtually exclusively employed in corporate settings. Since April 2003, when Windows Server 2003 was released,

Microsoft has marketed Windows Server under this brand. Server versions of Windows were available even before this. Windows NT 4.0, for example, was offered in both desktop (general purpose) and server variants. Normal users generally never need to be concerned about Windows Server. We won't locate it on store shelves or download it by mistake from Microsoft when we're looking for the ordinary version of Windows.

1.6.2 Operating System similarities between Windows and Windows Server

If we install a fresh copy of Windows 10 and Windows Server 2016, it's possible to get the 2 confused at first. They all could share the same desktop, start button, and even task view button. They share the same kernel and may be able to execute the same applications. On both, we can install Google Chrome or Microsoft Office, for example.

But that's where the similarities end. Microsoft developed Windows 10 for use as a desktop that users sit in front of, and Windows Server for use as a server that runs services that users access across a network (it's right there in the name). While Windows Server has a desktop option, Microsoft recommends installing the server without a graphical user interface (or uninstalling it entirely), leaving only a command line to operate with, which decreases the server's cost. This includes a push to install Nano Server, which neglects the GUI and local login capabilities in favor of taking up significantly less space than the normal Server installation.

1.6.3 Operating System differences between Windows and Windows Server

For quite some time, Windows has been the most popular operating system in the world. Microsoft releases a new version of Windows every couple of years with enhanced security features, a more user-friendly interface, and newer software in generally. Microsoft does not, however, provide only one type of Windows OS. There's also a server line, with Windows Server 2019 being the most recent release. I'll compare and contrast these two systems. The primary distinction between Windows 10 and Windows Server is that Windows 10 is designed for use on personal computers at home, whereas Windows Server is a dedicated server operating system.

When we enable the graphical user interface in Windows Server, a server manager appears, which is a clear distinction between the two (Windows and Windows Server). We may add functionality like DHCP services, Windows Deployment services, and more to the server manager. These extra capabilities assist in the remote deployment of an operating system, the creation of static IP addresses for client devices, the formation of domain users, and more. All of these functions aren't included with Windows 10, but they can be enabled using third-party software.

As mentioned above Windows Server has server-specific tools and software that aren't available in Windows 10. Windows PowerShell and Windows Command Prompt, for example, are pre-installed in the operating system to allow you to handle your operations remotely. Additionally, Windows Server may support a variety of business-friendly software, such as Active Directory and DHCP, that is developed especially for servers. While some of these tools are compatible with Windows 10, others may necessitate the use of third-party applications. Windows Server, on the other hand, misses several of the more 'fun' capabilities seen in Windows 10. Because the system is intended for commercial usage, it lacks consumer features such as Edge, the Microsoft Store, and Cortana.

And also, Windows Server includes SMB Direct, which speeds up file sharing, as well as RFS (Resilient File System) compatibility. We must utilize Windows 10 Pro for workstations to gain the same functionalities without having to install Server on Windows.

Since Windows Server is designed for businesses, it comes with a lot of business software. The following are some of the functions that these tools can help a server fulfill.

- Active Directory : Active Directory is a user management system that enables a server to function as a domain controller. The domain system manages all user account authentication instead of each user connecting into a local workstation.
- DHCP : The Dynamic Host Configuration Protocol (DHCP) allows a server to assign IP addresses to all network devices automatically. This is most likely handled by the router at home. In a business setting, however, IT professionals can take advantage of Windows Server's enhanced DHCP functionality.
- File and Storage : Another common application for a server is as a file server for the company. This allows us to keep vital data in one place and regulate who has access to which files by setting permissions.
- Print Services : It's a waste of time for IT employees to configure each printer manually for each new workstation if a company has dozens of printers around the premises. We can easily map printers to computers and reduce redundant effort by setting up a print server.
- Windows Update Services : Businesses frequently do not want all Windows updates to be installed at the same time. We may route all workstation updates through that server and define specific rules for how they should work by setting up a server as a Windows Update controller.

These are just a few of the server roles supported by Windows Server. Typically, a corporation will have numerous servers and distribute the above duties among them. These features are not included in standard Windows installations. Some of this functionality can be duplicated with third-party tools, but they won't be as powerful.

Power user capabilities like the Command Prompt and other administrative tools are retained in Windows Server, as expected. Server editions of Windows, on the other hand, lack many of the quality-of-life features seen in Windows 10. For example, the Microsoft Store, Cortana, and other new Windows 10 features will not be available in Windows Server 2016 and 2019. It doesn't include programs like Our Phone, and we have to get Microsoft Edge individually even on Windows Server 2019. We can't log in using a

Microsoft account on Windows Server Operating systems, either. We don't need these consumer-facing capabilities on a server OS as they're built for enterprise use.

Some apps also check to verify if we're running Windows Server before installing. The software may not work on a server version of Windows in some instances. By default, Windows Server is considerably more restricted. The default browser is still Internet Explorer, but the security settings are far more restricted than usual. This makes sense because compromising a server is far more serious than compromising a single desktop.

We'll need a graphical user interface (GUI) to explore Windows 10, which is an interface on our device that allows us to navigate the operating system. In the case of Windows Server, however, this is not the case. While a GUI is not required by the server operating system, it is always an option. The product can be installed in two ways. Server Core or Desktop Experience are the two options. We can install only the server roles that we require with Windows Server. This gives the user more flexibility, allowing us to run our operations in whatever way suits us best, while also reducing the Windows Server footprint.

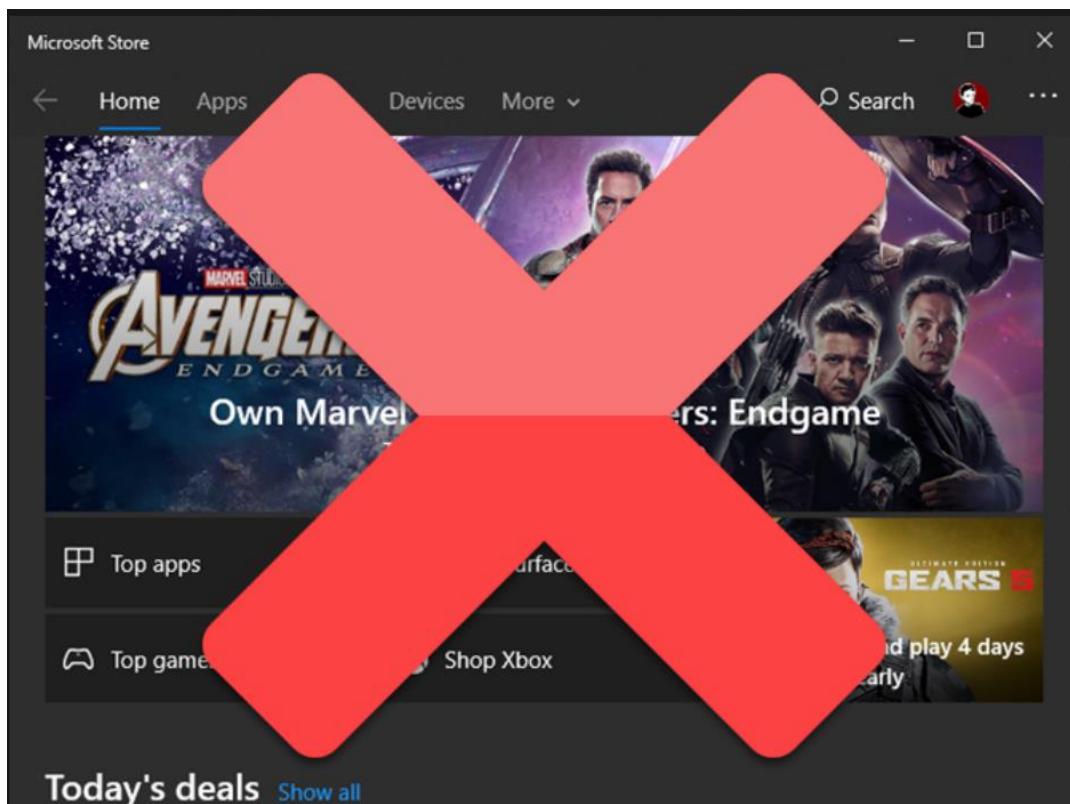


Figure 1. 45 Microsoft Store isn't in Server Software

1.6.4 Hardware differences between Desktop PC and Server

The majority of people are unconcerned about the maximum quantity of RAM that can be installed in their machine. With a 64-bit version of Windows 10 Pro, we can install up to 2TB of RAM. However, the majority of customers' PCs have no more than 32GB of RAM, therefore installing 1TB of RAM is out of the question. However, Windows Server can handle up to 24TB of RAM, which really is incredible. It also allows us to use up to 64 CPU sockets, which is far more than Windows 10 Pro's two sockets.

For example, a server hosting dozens of virtual machines need a lot of RAMS to keep them all working smoothly at the same time. This highlights another crucial aspect of Windows Server: it does not always run-on physical hardware. Some firms buy a few real servers and then use Windows Server to operate multiple virtual machines to accomplish various functions, such as those mentioned above.



Figure 1. 46 Server Hardware

Table 1. 11 Windows 10 Hardware requirements against with Windows Server Hardware requirements

| Windows 10 Hardware requirements | Windows Server Hardware requirements |
|---|---|
| The highest amount of RAM that Windows 10 Pro could have is 2 TB. | Windows Server can support up to 24 TB of RAM. |
| A standard computer user will not require 2 TB of RAM. | More RAM, on the other hand, means more capacity for the server. As a result, a server with sufficient RAM can easily accommodate users, VMs (Virtual Machines), and computers. |
| The maximum number of sockets in Windows 10 is 2. | Server 2016, on the other hand, might have up to 64 sockets. |
| 32-bit Windows 10 could only manage 32 cores, whereas 64-bit Windows 10 could handle 256 cores. | The number of cores in Windows Server is limitless. (We'll need Windows 10 Pro for Workstations, which comes with 6TB RAM and 4 CPUs, to get a little closer to using such huge capabilities.) |
| Desktop CPUs:- Intel i3, i5, i7, i9 | Server CPUs:- Intel Xeon , Intel Itanium |

1.6.5 Environment differences between Desktop PC and Server

It receives Windows updates more quickly and very seldom. Cortana and Timeline are features that aren't available on Windows Server. Installing Windows software, establishing preferences, and transferring preferences are all simple tasks with Microsoft Account. Other features in Windows 10 include Windows Subsystem, Progressive Web Apps, and Our Phone for Linux. These functions are heavily dependent on Microsoft Store, which Windows Server lacks.

When we boot up Windows Server, it starts up first, then a server manager appears, which is a clear distinction between Windows and Windows Server. We may add features like DHCP services, Windows Deployment services, and more to the server manager. These extra capabilities assist in the remote deployment of an operating system, the creation of static IP addresses for client devices, the formation of domain users, and more. All of these functions aren't included with Windows 10, but they can be enabled using third-party software. Windows Server includes SMB Direct, which speeds up file sharing, as well as RFS (Resilient File System) compatibility. We must utilize Windows 10 Pro for workstations to gain the same functionalities without having to install Server on Windows.

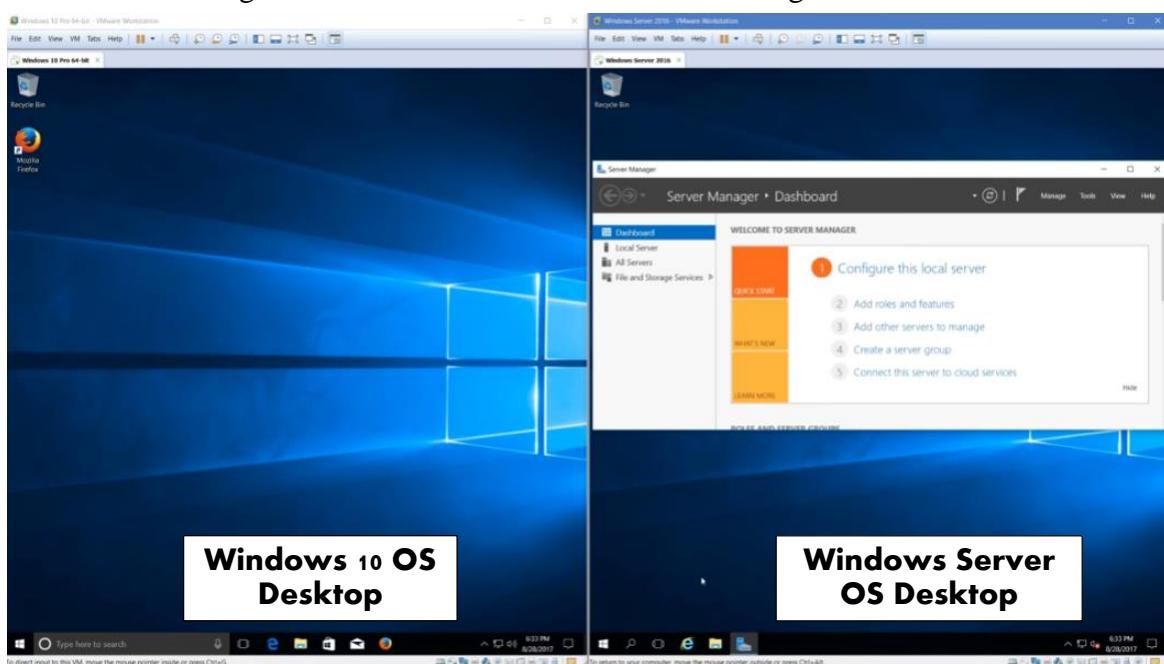


Figure 1. 47 Windows10 OS desktop difference with Windows Server OS desktop

1.6.6 Server options limitations

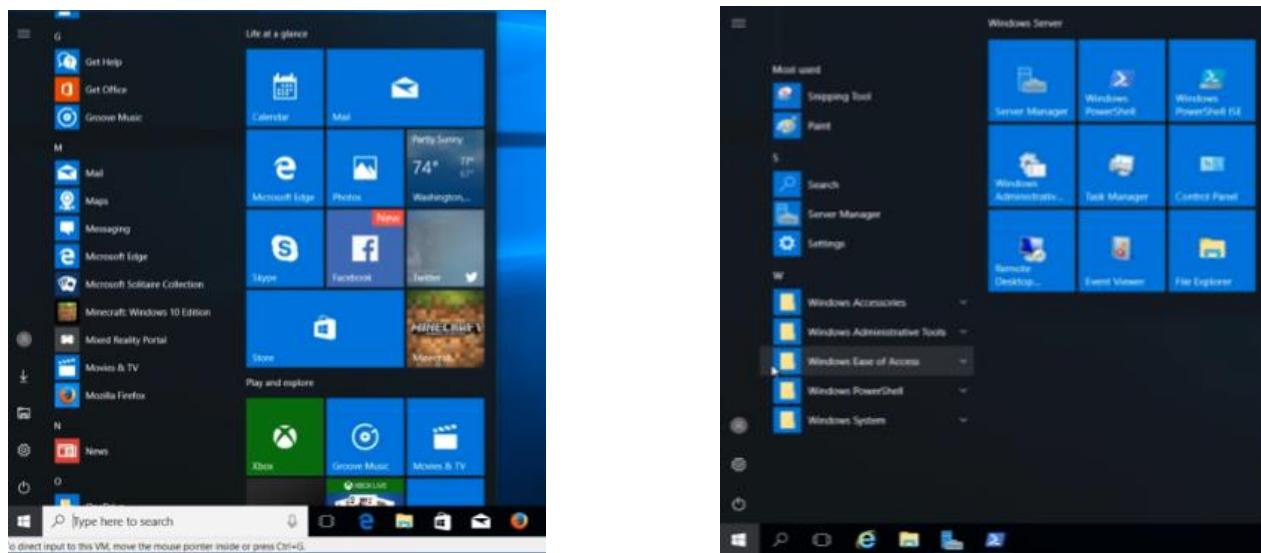


Figure 1. 48 Windows10 OS windows key difference with Windows Server OS desktop

On Windows Server, we will not find Edge, Cortana, or the Microsoft Store. In contrast to Windows 10, Windows Server employs IE (Internet Explorer), which cannot be used for web browsing. If we want Google Chrome to work, we must make exceptions for all Google URLs in order to complete the download. While browsing any website in Internet Explorer, we can see Windows Server's enhanced security.

Also, we are unable to enter into our Microsoft accounts on Windows Server, resulting in reduced accessibility (we are unable to transfer settings from another PC to it). We can only use a domain account to log in. Using group policy on Windows Server, we may entirely prevent updates. Windows 10 will, however, acquire the ability to pause updates.

1.6.7 Price differences between Windows and Windows Server

Windows Server is not cheap, as one might expect from a business-oriented product. It's significantly more expensive than a consumer version of Windows, and it comes in a variety of flavors to suit the demands. The pricing webpage for Microsoft's Windows Server 2019 offers us an estimate of what we might pay for the server OS. We will also need to pay for CALs (Client Access Licenses) to utilize the services lawfully, depending on how many users will access the server.

Windows Server was traditionally deployed on a physical onsite server, which has far more hardware capabilities than a desktop, as noted above. However, we may also use a cloud service such as Microsoft Azure to operate Windows Server. This frees us up to transfer the maintenance of a physical server to a cloud provider like Microsoft. Furthermore, rather than paying for a new physical server all at once, companies can spread the cost of upgrading out over a subscription. What works best is determined by the demands of the organization.

The following are the prices for Windows Server in 2019:

| | |
|------------|---------|
| Datacenter | \$6,155 |
| Standard | \$972 |
| Essentials | \$501 |

Figure 1. 49 Prices for Windows Server in 2019

The following are the prices for Windows 10:

| | |
|---------------------------------|----------|
| Windows 10 Home | \$139 |
| Windows 10 Pro | \$199.99 |
| Windows 10 Pro for Workstations | \$309 |

Figure 1. 50 Prices for Windows 10

No matter how large or little my Hospital Management System is, if I want to have an online presence, I'll need to think about web servers. Other solutions will be available to me, but they may not offer the same level of flexibility and dependability as web servers. Since we have a variety of web servers to select from, choose which one will host my system can be difficult. One of the key reasons for the large number of servers available is because of this. Each server will serve a distinct role. I should create a list of my client's aims and needs in order to ensure that my system is hosted on the server that is best suited to their demands.

By developing a list of customer needs, we will most likely be able to offer ourselves an estimate of what type of performance we can expect from our system. I can investigate systems and hardware to see if they will satisfy the basic specifications I've set once I've decided on the system's requirements and needs. I'll also have to figure out which support systems and prices are the most cost-effective for my system. When I wish to use a server, there are a few things I need to think about. If I intend to handle a significant burden on a daily basis, I'll need a good server that can handle my projects.

When using a server main thing we want to focus on the speed. If my Hospital Management System website gets too long to load, I may be losing users, and it's time to start thinking about whether or not we're utilizing the correct web server. According to Nielsen Norman Group research, we only have ten seconds to convey our value proposition. Visitors will most likely navigate away if they aren't hooked early on. On mobile devices, we have even less time; according to Google study, 53% of mobile consumers will leave a website if it does not load in 3 seconds. Furthermore, according to the Stanford Persuasive Technology Lab, the simplicity of use of our website affects its trustworthiness, which may affect whether or not an user does business with us.

1.6.8 Differences between on-site servers and cloud servers

Table 1. 12 Differences between on-site servers and cloud servers

| On Site Servers | Cloud Servers |
|--|---|
| High initial hardware investment (or ongoing cost if rented) plus installation. | The server itself is free, but we must pay continuous hosting expenses. |
| Our data is within our physical control. | Access to our data is controlled by our cloud provider. |
| We can create a server that is suited to our company's requirements. We'd have to upgrade or replace the equipment as these requirements change. | Our cloud services' quality of service (storage space, data speed, and so on) may be adjusted as needed. Our requirements may exceed our financial resources. |
| To maintain the hardware and maybe install and upgrade software, we'll need in-house IT workers or an IT support provider. | Our payments cover the costs of maintaining hardware and installing software, but different providers offer different levels of tech support. |
| We are not cut off from your data due to an internet outage. | Unless we have redundant internet connections, an internet outage could cut us off from our data. |
| Data that isn't backed up offsite can be lost if our hardware is damaged. | Data stored in the cloud is unaffected by physical damage to hardware. Cloud providers, on the other hand, have different levels of backup capabilities and accountability. We may need to make additional backup plans. Also, data restoration can be costly, especially if we need it right away. |

1.7 Choosing a Server type for Hospital Management System

Because of that above explanation about differences between on-site servers and cloud servers, my Hospital Management System may be able to cost-effectively minimize the weaknesses of on-site services with the strengths of cloud solutions and minimize the weaknesses of cloud solutions with the strengths of on-site services. Hybrid in-house/cloud models are rapidly evolving. It's critical that we don't commit our company to any long-term agreements without first consulting with IT professionals I trust.

*To limit third-party and hacker access, we need keep our most sensitive client data or proprietary software on on-site servers. However, if we do this, we must ensure that our backup procedure works and is frequently tested. We'd also need IT knowledge, either on staff or through a third-party vendor.

*Subscriptions to the cloud can be used for significant applications that might otherwise drag down our computers and systems. Major software suites such as Microsoft Office 365 and Adobe Creative Cloud have become cloud-based subscription services.

*We utilize a cloud-based backup system in addition to backing up our data from an in-house server to a different in-house storage device. Even if our physical assets are damaged or destroyed, this protects us from long-term business interruption.

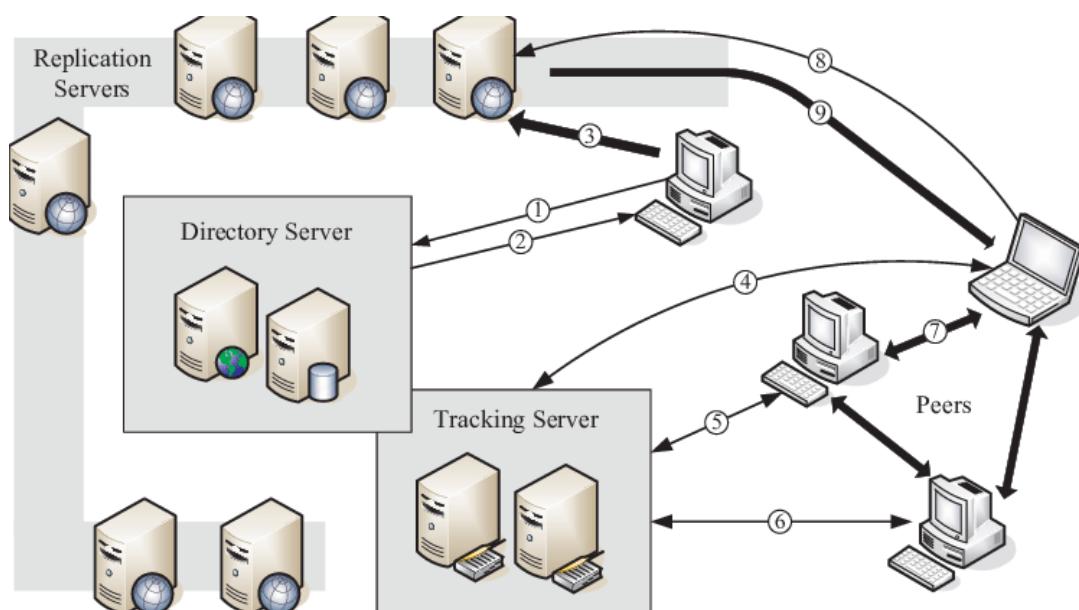


Figure 1. 51 Hybrid Server

1.8 DNS and DNS Server

1.8.1 Introduction

Computers do not use names in the same way that humans do. They use IP addresses, which are numbers. As a result, network engineers created DNS protocol to bridge the communication gap between computers and humans. So, DNS stands for Domain Name System. DNS is a service that converts domain names into IP addresses. In other words, DNS allows people to find websites by using human-readable hostnames rather than numeric IP addresses.

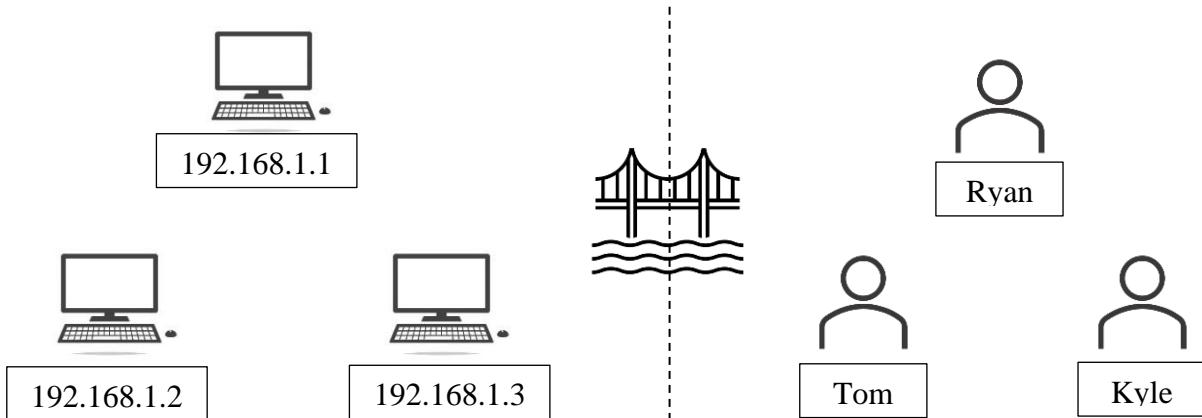


Figure 1. 52 How DNS bridge the communication gap between computers and humans

For example, 142.250.191.142 this is “google .com” IP address. Since we have DNS, we won't have to type this IP address into the address box. Simply input google.com, and the DNS Server will look for the IP address for that domain name in its database. So basically, DNS works like a phonebook. We don't have to keep track of tens of thousands of IP addresses. The only thing we need to remember is the domain name. It's far easier for us to enter in a URL than it is to type in IPv4 IP addresses (for example, 192.168.1.1) or, even worse, IPv6 addresses (E.g. 2400:cb00:2048:1::c629:d7a2). Hence, DNS servers, which are machines specialized to responding DNS queries, are responsible for all of this.

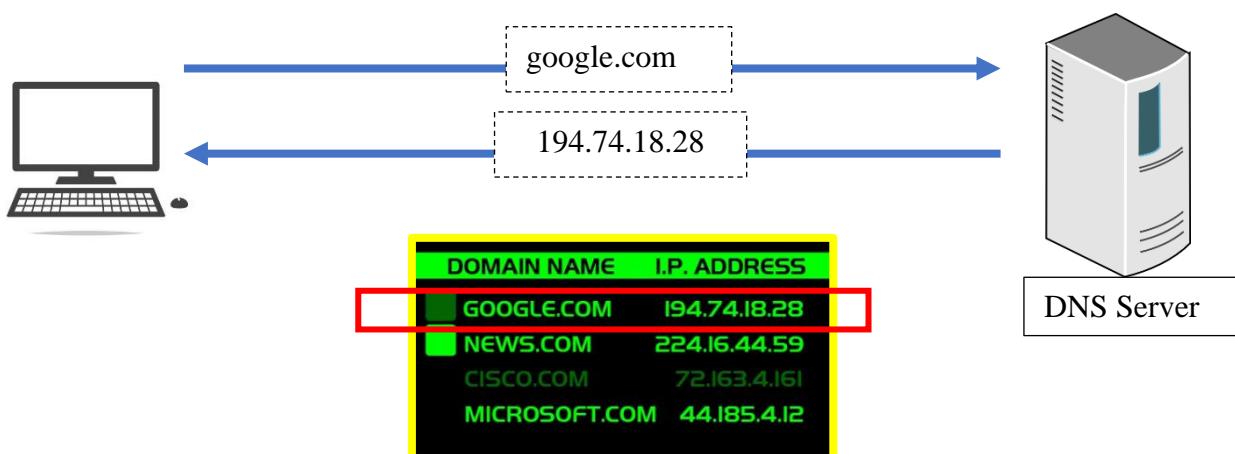


Figure 1. 53 How DNS works like a phonebook

(PowerCert Animated Videos, 2016)

1.8.2 How DNS works

DNS resolution is the process of translating a hostname (for example, www.example.com) into a computer-friendly IP address (such as 192.168.1.1). Each device on the Internet is assigned an IP address, which is used to locate the proper Internet device in the same way that a street address is used to locate a specific residence. When a user requests a webpage, a translation must take place between the user's input (example.com) and the machine-friendly address required to locate the example.com webpage.

To comprehend the DNS resolution process, it's necessary to first learn about several hardware components that a DNS query must transit through. Apart from the initial request, the DNS lookup occurs "behind the scenes" for the web browser and requires no involvement from the user's computer.

(What is DNS? | How DNS works, n.d.)

1.8.3 Process of viewing a Website

When we type a domain name into a web browser, our computer resolves the hostname, such as www.yahoo.com. Our computer will then search its local DNS cache for the IP address associated with the domain name. This cache keeps track of the data that our computer has lately saved. The webpage will be displayed if it is available locally. If our computer doesn't have the information, it will run a DNS query to get it.

For example, when we input "yahoo.com" into a web browser, it searches its cache memory for the "yahoo.com" IP address. If it can't, the inquiry will be sent to the Resolver Server. The Resolver Server is simply our Internet Service Provider (Internet Service Provider). It will look for the IP address "yahoo.com" in its cache memory. If it is unable to do so, the query will be sent to the Root Server.

Root server is basically root of the DNS hierarchy. The query will be sent through Resolver Server to the TLD Server (Top Domain Level Server). Root server doesn't know the IP addresses. Root server does know where to send the query. The query will be sent through Resolver Server to the TLD Server (Top Domain Level Server).

The information for the Top-Level Domain is stored in TLD. as an example (.com , .net , .org). Since Yahoo has a ".com" domain, it is managed by a TLD server. But TLD doesn't know the IP addresses. It will use Resolver Server to route the query to the Authoritative Name Servers.

The Authoritative Name Servers are in charge of keeping track of all domain IP addresses. The query then moves up the authoritative DNS server chain. The search will go on until a nameserver for the domain is found since these authoritative name servers are in charge of maintaining these records for their domain names. Then the IP address of "yahoo.com" will be forwarded to the Resolver Server. Finally, Resolver Server will provide the "yahoo.com" IP address to our PC.

Once Resolver Server Retrieve the IP address, it will store it in its cache memory in case it receives another query for yahoo.com. So, it doesn't have to go through all those steps again. When simply comprehend this process, we obtained our IP address from the DNS record and sends it to our browser. The full lookup operation takes milliseconds to complete from start to finish.

(Singer, 2021)

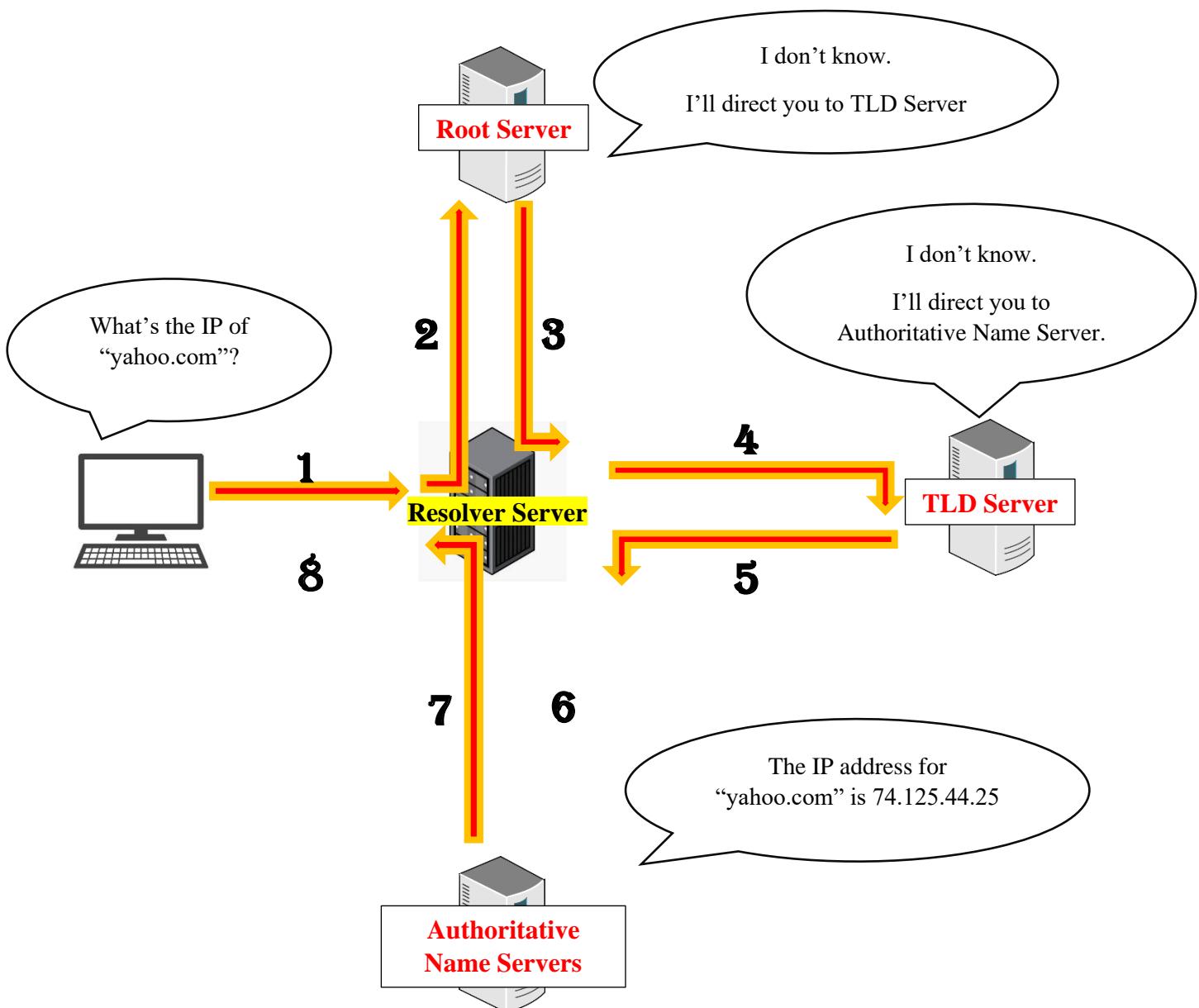


Figure 1. 54 Process of viewing a Website

1.9 DNS Hierarchy

1.9.1 DNS Hierarchy with Top Level Domains

DNS manages its distributed database system through a hierarchy. The DNS hierarchy, commonly known as the domain name space, is similar to eDirectory in that it is an upside-down tree structure. The root domain is the single domain at the top of the DNS tree structure. The root domain is denoted by a period or a dot (.). The top-level domains, which are below the root domain, divide the DNS hierarchy into segments. The top-level DNS domains and the types of organizations that use them are listed below. The domain name space is further divided into subdomains that represent individual organizations below the top-level domains.

Table 1. 13 Top-Level DNS Domains with examples

| Domain | Used By |
|--------|---|
| .com | This is used by commercial websites. Ex:- facebook.com |
| .edu | This is used by educational websites. Ex:- ucla.edu |
| .gov | This is used by governmental websites. Ex:- whitehouse.gov |
| .org | This is used by organizational websites. Ex:- redcross.org |
| .net | This is used by network organizations. Ex:- nsf.net |
| .int | This is used by international organizations. Ex:- nato.int |

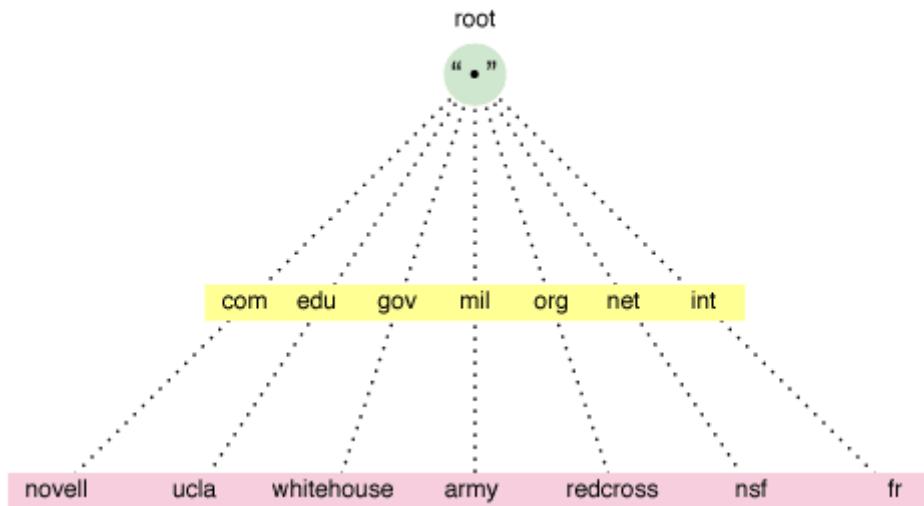


Figure 1.55 DNS Hierarchy

1.9.2 Domains and Subdomains in DNS Hierarchy

In the DNS hierarchy, a domain is a label. A domain is represented by each node in the DNS tree. Individual organizations or entities are represented by domains beneath the top-level domains. These domains can be further subdivided into subdomains to facilitate administration of an organization's host systems.

For example, under the ".com" top-level domain, Company A creates a domain called "companya.com." Company A's Chicago, Washington, and Providence offices each have their own LAN. As a result, Company A's network administrator decides to build a different subdomain for each division, as indicated in the Domains and Subdomains picture below. A subtree's domains are considered part of the domains above it. As a result, "chicago.companya.com" is a subdomain of companya.com, and both are in the ".com" domain.

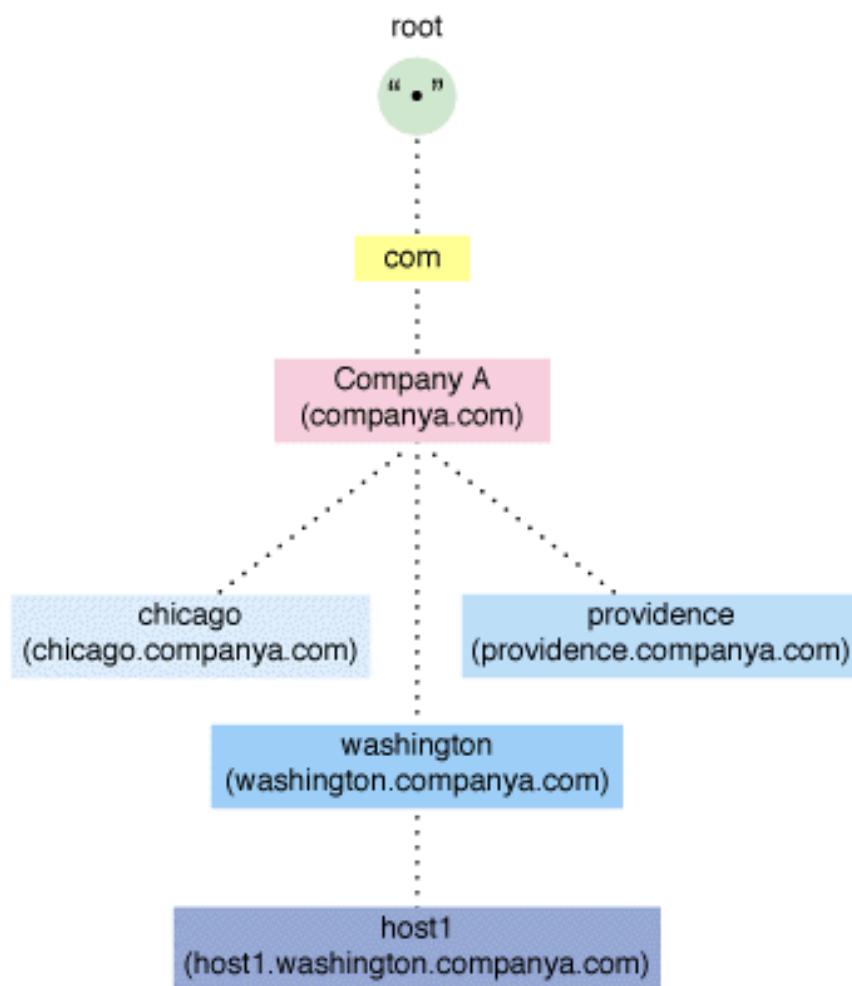


Figure 1. 56 Domains and Subdomains in DNS Hierarchy

1.10 DNS Types

1.10.1 Types of DNS and uses of them

All DNS servers are classified into one of four groups.

1. Recursive Resolvers
2. Root Nameservers
3. TLD Nameservers
4. Authoritative Nameserver

In a normal DNS lookup (without caching), these four DNS servers collaborate to send the IP address for a particular domain to the client.

1) Recursive Resolvers

The first stop in a DNS query is a recursive resolver (also known as a DNS recursor). Between a client and a DNS nameserver, the recursive resolver serves as a middleman. A recursive resolver will either respond with cached data or send a request to a root nameserver, which is followed by another request to a TLD nameserver, then finally a request to an authoritative nameserver after receiving a DNS query from a web client. The recursive resolver sends a response to the client after getting a response from the authoritative nameserver with the requested IP address.

The recursive resolver would cache data collected from authoritative name servers during this procedure. When a client requests the IP address of a domain name that has recently been requested by another client, the resolver can skip the nameserver communication and just deliver the requested record from its cache to the client. The majority of internet users utilize their ISP's recursive resolver, however there are alternative options.

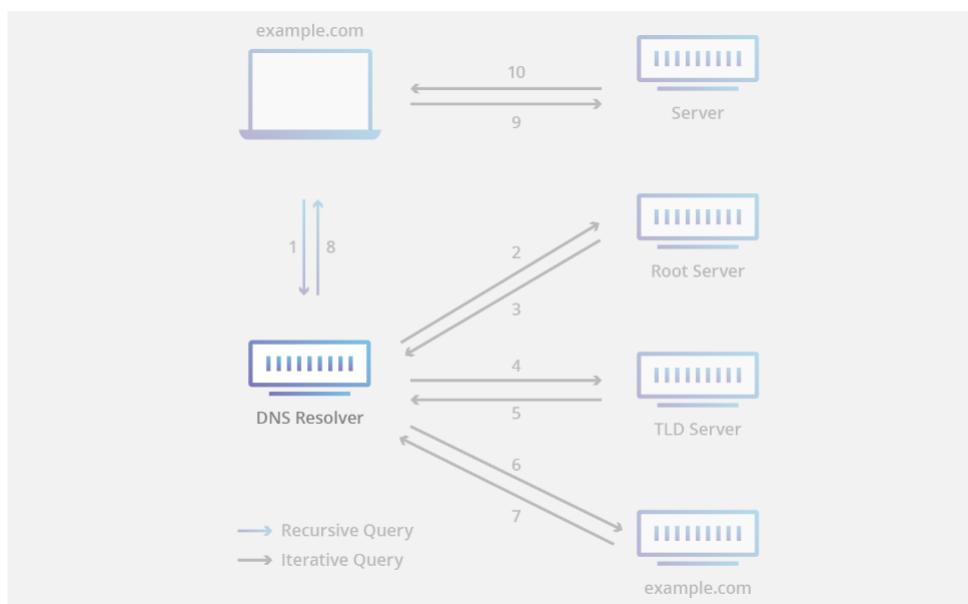


Figure 1. 57 Recursive resolver

2) Root Nameservers

Every recursive resolver is aware of the 13 DNS root nameservers, which are the first stop in a recursive resolver's search for DNS records. A root server takes a domain name query from a recursive resolver, and the root nameserver responds by sending the recursive resolver to a TLD nameserver based on the domain's extension (.com, .net, .org, etc.). The Internet Corporation for Assigned Names and Numbers (ICANN) is in charge of the root nameservers.

While there are 13 root nameservers, the root nameserver system does not consist of only 13 machines. There are 13 different types of root nameservers, but each one has several copies throughout the world that employ Anycast routing to offer fast replies. There are 632 distinct servers if we add up all the instances of root nameservers (as of October 2016).

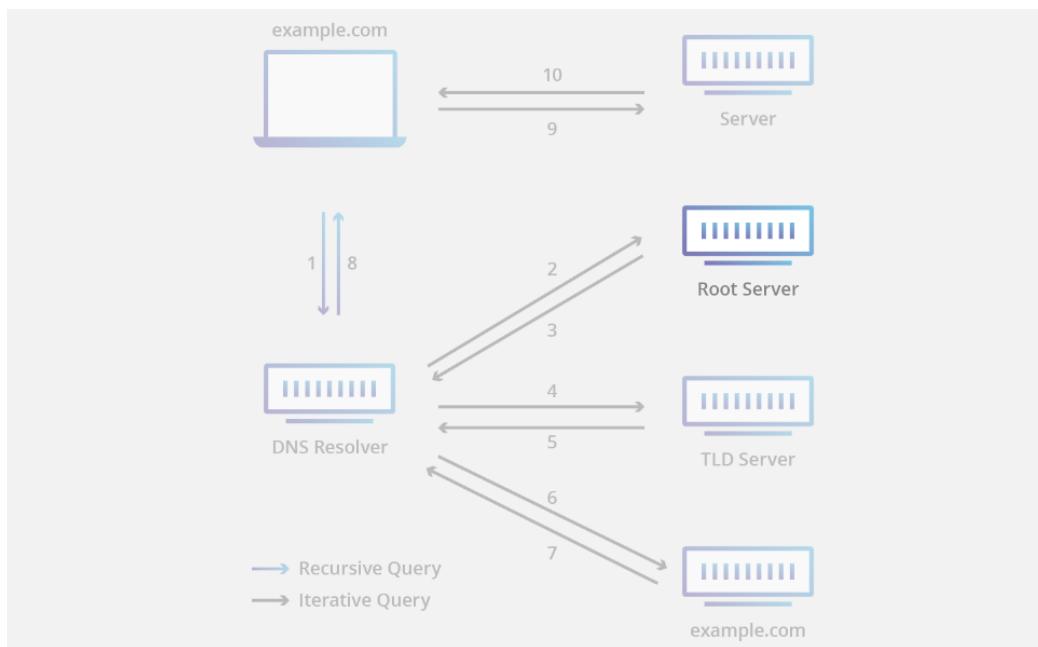


Figure 1. 58 Root nameserver

3) TLD Nameservers

A TLD nameserver keeps track of all domain names with the same domain extension, such as .com, .net, or whatever comes after the last dot in a URL. A .com TLD nameserver, for example, includes information for each and every website that ends in '.com.' If a user searched for google.com, the recursive resolver would send a query to a .com TLD nameserver, which then would reply by pointing to the authoritative nameserver for that domain.

The Internet Assigned Numbers Authority (IANA), a division of ICANN, is in charge of TLD nameserver management. The TLD servers are divided into two groups by the IANA:

- Generic top-level domains : These are non-country-specific domains. Some of the most well-known generic TLDs are .com, .org, .net, .edu, and .gov.
- Country code top-level domains : Any domains that are specific to a country or state fall under this category. .uk, .us, .ru and .jp are some examples.

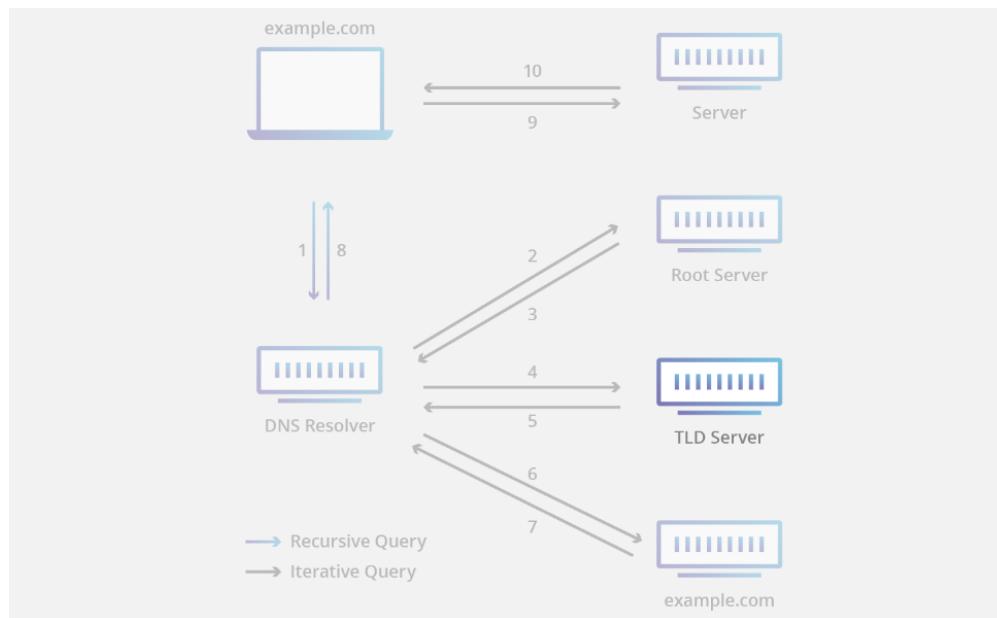


Figure 1. 59 TLD nameserver

4) Authoritative Nameserver

A recursive resolver will be directed to an authoritative nameserver when it receives a response from a TLD nameserver. The resolver's final stage in the path for an IP address is usually the authoritative nameserver.

The authoritative nameserver stores information that is specific to the domain name it is responsible for (e.g. google.com). It can also provide a recursive resolver with the server's IP address from the DNS A record. Otherwise, if the domain has a CNAME record (alias), it will provide an alias domain to the recursive resolver, forcing the recursive resolver to execute a fresh DNS query to obtain a record from an authoritative nameserver (often an A record containing an IP address).

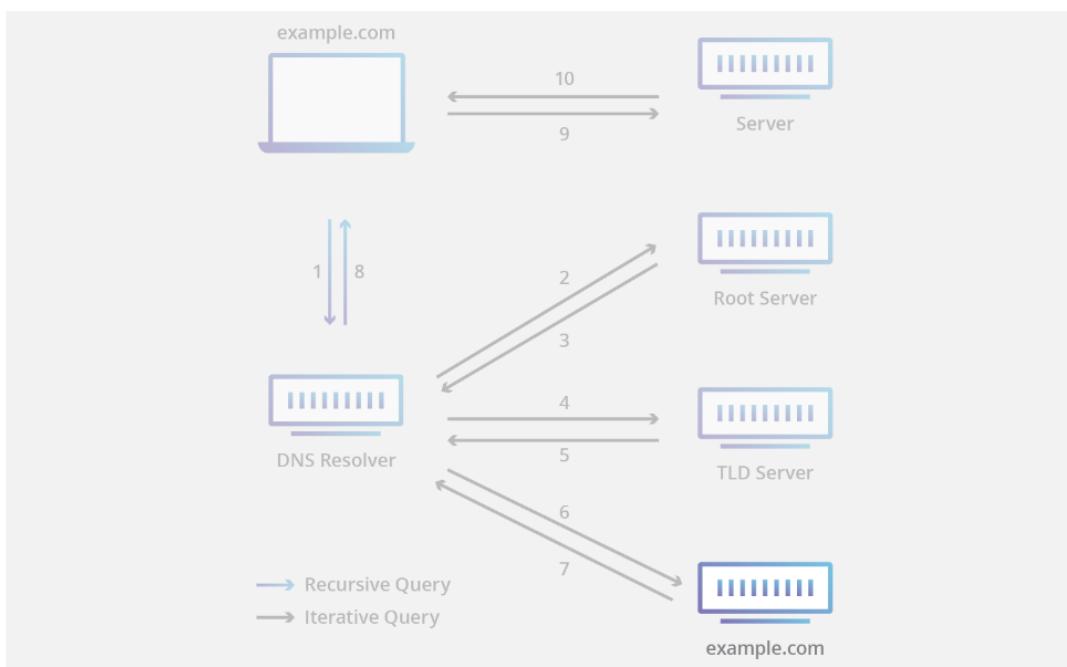


Figure 1. 60 Authoritative nameserver

1.11 What is URL and Parts of URL

If our website is built like a house, the URL is similar to the address of that house. It helps our visitors discover our website by defining where it lives online, similar to how our house address determines where we live in a community. A URL is made up of five parts: the scheme, the subdomain, the top-level domain, the second-level domain, and the subdirectory.



Figure 1. 61 Parts of URL

Scheme →

When a web server views a page on our website, the scheme instructs it which protocol to utilize.

`https://blog.hubspot.com/marketing/parts-url`

Figure 1. 62 Parts of URL - Scheme

Subdomain →

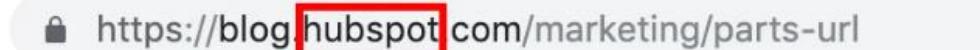
If our website is a house, our subdomains are individual rooms within it. A subdomain in a URL specifies which page of our website should be served by the web browser.

`https://blog.hubspot.com/marketing/parts-url`

Figure 1. 63 Parts of URL - Subdomain

Second-Level Domain →

The name of the website is the second-level domain. It lets people know they're on the website of a specific organization.

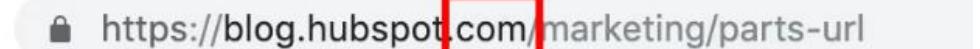


https://blog.hubspot.com/marketing/parts-url

Figure 1. 64 Part of URL -Second Level Domain

Top-Level Domain →

The top-level domain identifies the type of entity that our company is on the internet.

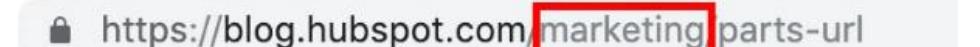


https://blog.hubspot.com/marketing/parts-url

Figure 1. 65 Parts of URL -Top Level Domain

Subdirectory →

A subfolder is also known as a subdirectory. This assists both people and web crawlers in determining which section of a webpage they are on.



https://blog.hubspot.com/marketing/parts-url

Figure 1. 66 Parts of URL -Subdirectory

(Chi, 2021)

1.12 Search Engine Optimization (SEO) optimization

1.12.1 Introduction

SEO optimization, or simply SEO, is a set of techniques for assisting search engines in determining what kind of content is available on a website and, as a result, assisting users in finding that information. 'Search engine optimization' is the acronym for 'search engine optimization.'

In reality, SEO simply comes down to creating high-quality content and then optimizing it by matching the text with how users search for that type of information. SEO-optimized sites include clear and direct titles, useful section headers, detailed file names and alt tags for pictures, and easy-to-understand site hierarchies, and are much more than merely stuffing material full of keywords.

From the viewpoint of a search engine, it's essential to make it simple for visitors to obtain the information they require. Hence the user experience of a website is crucial. The user experience is influenced by site performance; waiting for pages to load is uncomfortable for the user.

Keyword research is the practice of adding keywords in an unusually high number of places in text in order to get an advantage in search engine rankings. Google no longer considers keyword frequency to be relevant and sometimes penalizing sites for obvious cases of keyword stuffing.

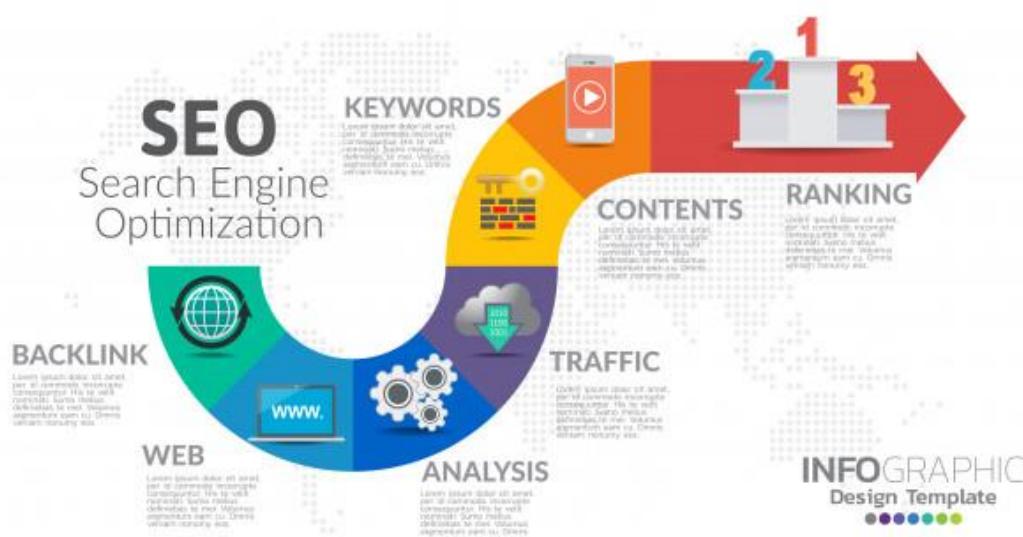


Figure 1. 67 Search Engine Optimization (SEO) optimization

1.12.2 Search Engine rankings

When a person uses a search engine to do a search, the search engine generates a list of websites that are ranked from most likely to least likely to satisfy the query. This list is usually separated into a series of search engine results pages (or SERPs) that list websites that may answer the user's query, with 10 results on each SERP. The majority of people will not look past the first page of search results, preferring to narrow their query and try again. As a result, most SEO experts aim to place the website they're working on in the top ten search results for the keywords they're targeting.

A website's position in search engine results is determined by numerous of factors, including how relevant it is to the query and how trustworthy the search algorithm deems to be. By optimizing a website for search, website owners ensure that their content appears as high in search rankings as possible, increasing the likelihood that consumers will click.

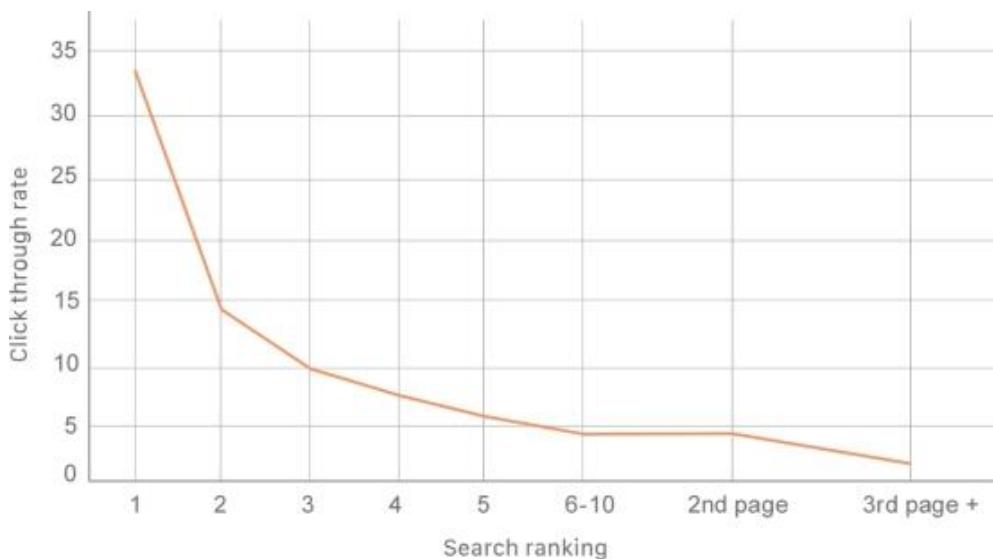


Figure 1. 68 Search-ranking-click-through-rates

Google controls the search engine industry in the West, and its algorithms are largely responsible for SEO best practices. In North America, Google accounts for about 90% of all searches, with Bing and Yahoo Search accounting for the majority of the remaining market. Other search engines, on the other hand, maintain influence in other regions of the world; in China, for example, Baidu is the most used search engine.



Figure 1. 69 Top 10 search engines

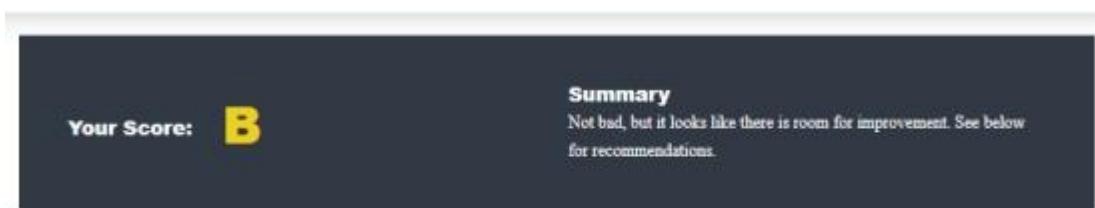
1.12.3 Improving a site's index value and rank of the Hospital Management System (HMS) through search engine optimization

Google said in 2010 that site speed would be taken into account when determining search rankings. Since then, site performance has become increasingly important, and optimizing for performance is now an important aspect of SEO optimization. As a result, code has a significant impact on the site's performance, particularly in terms of loading time. For faster load times in the Hospital Management System (HMS) code, I make sure to delete irrelevant characters, commas, spaces, and other unnecessary elements.

All users desire to work with a website that is of the highest quality possible. Using smaller pictures can help improve page speed in this case. Finding a balance between image quality and image size is critical. Because of that, I used fewer images and less content on the webpage. And I used simple and clear keywords when creating the website. Google emphasized the user's perspective in their initial mention of site speed as a ranking criteria, emphasizing how a slow website negatively effects the user's experience. Users were also less inclined to stay on pages that loaded slowly, according to Google. (Users would be less inclined to stay long enough to get their question answered as a result of this.)

Since Google does not give information about its search algorithms, it's difficult to say how much site performance influences SEO. However, website usability and overall user experience are critical, particularly for mobile devices, with Google's own study revealing that 53% of mobile sites are abandoned entirely if they take more than three seconds to load.

While some rapid improvements may have a good impact on the loading time of my Hospital Management System (HMS), it's critical to first obtain accurate page speed data. Running a page speed test is the best way to do this. When analyzing my web page speed, the "Pantheon Website Speed Test" is a wonderful place to start. All I had to do was type in the URL of the web page, the region for accuracy, and an email address. When I clicked "Start Speed Test," I was presented with page speed insights, which helped me get a better understanding of how my page was performing. The final report informed me of what was working well, what needed to be changed, and any additional information necessary to improve the website's loading speed. After performing a test on one site, here is how the report looked.



Results Breakdown

Your performance score takes is based on dozens of factors. Below, we display the most important and easily addressable, along with very rough estimates of goals to shoot for. We also compare your site to all sites on the Pantheon platform.



Page Details

The following results detail page size, number of requests and other important factors which affect performance. Pages over 2 megabytes, or require more than 30 HTTP requests to load are good candidates for review. Pantheon has detailed [guides](#) on how to optimize website performance.

| | | | |
|--------------------|--------|---------------|--------|
| PAGE TRANSFER SIZE | 1.8 MB | TOTAL SCRIPTS | 88 |
| PAGE CONTENT SIZE | 4.9 MB | AMP | NO |
| PAGE REQUESTS | 171 | RESPONSIVE | YES |
| HTTPS | YES | PERFORMANCE | 64/100 |

Figure 1. 70 Testing my HMS speed by “Pantheon” Website Speed Test

I can simply evaluate which parts of my site may be enhanced using these page speed insights. Rather than wasting effort and time debugging or guessing, this data allows me to pinpoint issue areas with pinpoint accuracy. I can simply evaluate which parts of my site may be enhanced using these page speed insights. Rather than wasting effort and time debugging or guessing, this data allows me to pinpoint issue areas with pinpoint accuracy.

Fortunately, there is a Google page speed test just for mobile searches. We don't even have to get up from our computers. All we have to do is go to "Test My Site" and give it a try. Then, in the search box, type domain and hit enter. After a few seconds, we'll have a thorough report with page speed insights and other useful data to assist us better understand our mobile site's performance. Remember that a site's optimal speed is three seconds or less. The findings will also include suggestions on how to enhance our mobile page speed SEO as well as industry comparisons to determine how our site compares to the competitors.

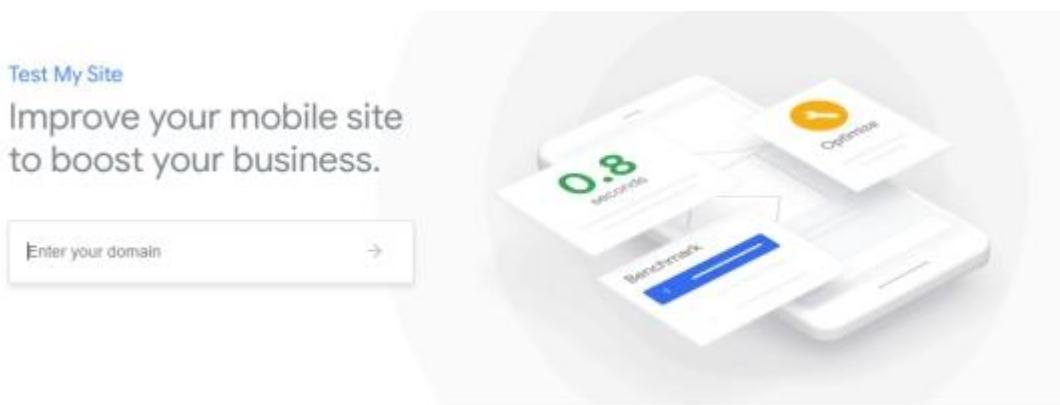


Figure 1. 71 Testing speed of a mobile site by “Test MY Site”

Another factor that affects SEO is indexing issues. Indexing issues may arise as a result of duplicate content from other websites, affecting our position and resulting in a lack of organic traffic from search engines. The Google algorithm might perform a thorough check on the site's content and compare it to similar sites. If it was copied from another website and isn't unique, it won't show up in a Google search result. It's usually caused by links from other sites that Google sees as duplicate content, even if I didn't intend for it to happen.

The canonical tags that appear on our site must be corrected to resolve this indexing issue. If the Hospital Management System content is unique and may cause an indexing issue on the site, the canonical tags should be removed. It will correct the website and may show up in search results.

1.13 SEO tools to use

Indexing difficulties are another aspect that has an impact on SEO. As a somewhat technical profession, SEO employs a variety of tools and software to assist in the optimization of webpages. The following are some of the most regularly used free and paid tools:

- Google Search Console : Google Search Console (previously known as "Google Webmaster Tools") is a free Google tool that is an essential component of every SEO's toolkit. GSC can help detect and resolve on-site technical issues by providing rankings and traffic information for top keywords and pages.
- Google Ads Keyword Planner : Another free tool supplied by Google as part of their Google Ads product is Keyword Planner. Despite the fact that it is created for sponsored search, it can be a useful tool for SEO because it provides keyword ideas and search volume, which can be useful when conducting keyword research.
- Backlink analysis tools : There are several link analysis tools available, the two most popular of which being Ahrefs and Majestic. Backlink analysis tools allow users to see which websites link to their own or competitors' websites, and they can also be used to find new links during link building.
- SEO platforms : Many different SEO platforms exist, each of which combines many of the tools that SEOs require to optimize websites. Moz, BrightEdge, Searchmetrics, and Linkdex are among the most popular. These tools keep track of keyword ranks, assist with keyword research, uncover on-page and off-page SEO opportunities, and do a variety of other SEO duties.
- Social Media : Although most social media sites have no direct impact on SEO, they can be a useful tool for networking with other webmasters and creating relationships that can lead to chances for link building and guest posting.

1.14 Web Development Technologies

1.14.1 Introduction

Every website developer should be familiar with a large number of rules and techniques related to web development. If we want a website to look and perform the way we want it to, we need to learn about web technologies that can help us get there. Knowing three key languages, JavaScript, CSS, and HTML, is often required to develop an app or a website. And, while it may appear difficult at first, understanding web technology and how it works becomes considerably easier once we know what we're doing.

Since computers are unable to speak with one another in the same manner that humans do, they must rely on codes. The markup languages and multimedia packages that computers use to communicate are known as web technologies. There are 2 types of web technologies.

Front-end technologies are used on a website or application's "client side." They're utilized to create any website's interactive elements and the components that users see and engage with. Colors and styles of text, pictures, buttons, and navigation menus are all part of this. Here followings are some common web technologies is being used.

1) HTML

Hyper Text Markup Language is the acronym for HTML. It is one of the most important technologies for web development. It offers the foundation for a web page's structure. HTML code assures that all of a website's content is formatted correctly. This is so that our browser can show the content correctly. A browser could not display text or load photos and other content without HTML.

HTML5, the most recent version of HTML, defines a significant variety of Application Programming Interfaces (API integration services) that can be used in combination with JavaScript to make a website more interactive and dynamic.

- Canvas: Canvas is an HTML5 element allows to create and modify pictures and shapes. More complicated cases, such as gaming visuals and animations, can also be handled with it.
- Web Storage: Web Storage is a technology that lets users to save information directly in their browser. Storing user login information and user preferences for a website are two examples of this.
- Service Workers: Service Workers enable a script to execute in the background whenever a web page is opened, and they are generally utilized in websites that have offline capabilities. It enables the use of web push notifications and makes pages available offline. Even if the browser isn't open, it can send these notifications.
- WebSockets: WebSockets enable a user and server to maintain a continuous two-way connection. Chats and notifications in web apps are the most common use cases.

2) CSS

The style and appearance of a web page are defined by Cascading Style Sheets, shortened as CSS. CSS specifies the design of a web page, while HTML specifies its structure. Page layouts, colors, fonts, and element placement are all examples of this. CSS is the skin of a web page, if HTML is its bones. It improves the appearance of the Internet, as well as our own website.

1.14.2 Server-side (back-end) technology

The "server side" of a website or application is where back-end technology come in. They're in charge of laying the technical foundations. They organize and store data and ensure that the front-end functions properly. For example, when a user enters login details to a site such as facebook, back-end technologies are utilized to validate those credentials. The server will give back the profile name, photo, and other relevant information once the credentials have been validated.

1) Programming Languages

Programming languages are techniques of communicating with computers and instructing them on what they should perform. They give a set of grammatical rules and a vocabulary for instructing a computer to accomplish specified tasks. There are numerous programming languages available, just as there are numerous human languages. Developers usually specialize in a small number of languages.

Here are some instances of several languages:

- **Java script :**This is Front-end language that make a website or application work. For several reasons, JavaScript is the most popular programming language among startups. It may be used for both front-end and back-end development. JavaScript is a very easy-to-learn language (although all languages have their own set of challenges), and it's often utilized in custom web application development. It's also being updated and extended with new features on a regular basis.
- **PHP :** This is a back-end programming language that allows a website or application to function. In web development, PHP is the most widely used language. PHP is the foundation of WordPress, a content management system (CMS) that powers 34% of the Internet. Having a better developer in any language is challenging, but choosing a popular one like PHP will provide you a broader pool of professionals to choose from and work with.
- **Python :** Another common programming language is Python. It's straightforward and employs a unique syntax that emphasizes readability. Python is useful for applications

and projects involving artificial intelligence (AI), statistics, data science, and machine learning. It's also possible to utilize it to make web applications.

- Java : Java is a widely used programming language in companies. It runs on a variety of systems, has detailed records, and is maintained by Oracle (a well-known software business) in the event of a problem. It's also possible to utilize it to create Android apps.
- Rust : Rust is a modern programming language that promotes safety, particularly safe concurrency. In software engineering, concurrency refers to a set of techniques that allow a computer program to complete many tasks at the same time. Safe concurrency allows us to do several activities efficiently while keeping our code safe and secure. Rust is a high-performing language that has achieved market momentum despite its age.

1.15 Web development frameworks

1.15.1 Introduction

Frameworks for web development are tools and libraries which developers are using to make developing in a specific language better and more efficient. They give interfaces to commonly utilized functionalities and also abstractions that make complex things easier to understand and manage.

They basically make the developing process more efficient because developers can draw from a pile of code resources when working. There's no need to start from scratch when frameworks provide all of the tools we need to create a website or application. Frameworks are less expensive than writing entirely new code since they give us a head start on our development. In comparison to writing brand new, untested code, they also reduce the development risk. There are frameworks for both the front-end and back-end development of websites.

1.15.2 Client-side (front-end) Framework

1) Angular

Google maintains the Angular front-end framework. It is the most comprehensive front-end framework, with the most combined functionalities, when compared to other front-end frameworks. Large companies and organizations that require a sophisticated front-end solution frequently utilize it due to its robustness and depth of functionality. It has a steep learning curve, like with other complicated applications.

One thing to keep in mind is that there are many versions of Angular, which might cause confusion. Because of the version difference, if we have an older version of Angular and wish to upgrade to a newer version for new functions and features, we need to rewrite a lot of our code.



Figure 1. 72 Angular Framework

2) React

Although React is more of a front end library than a framework, many developers consider it to be one. Component-based architecture was introduced for the first time in this framework. It comes with a virtual dom that is simple to use and speeds up dom manipulation. React was created by Facebook and may be used on both the server and client side.

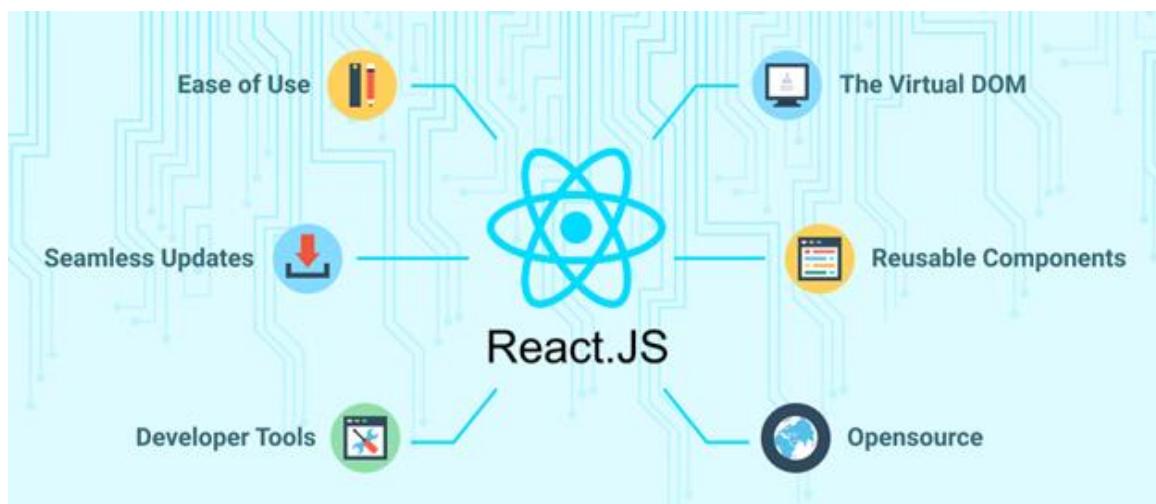


Figure 1. 73 React Framework

3) Vue

Vue.js, a new rising star, is a progressive framework that began as a side project. It focuses on component composition and declarative supplying and was created by Evan You. It is currently one of the most popular JS frameworks available. With its component architecture, users may create comprehensive frontend applications. It's one of the most adaptable frameworks on the market.

It allows us to work on current projects by taking a section of them and going forward in a seamless manner. It comes with a central library that focuses on the view layer. All of this makes it simple to integrate with other libraries and projects.

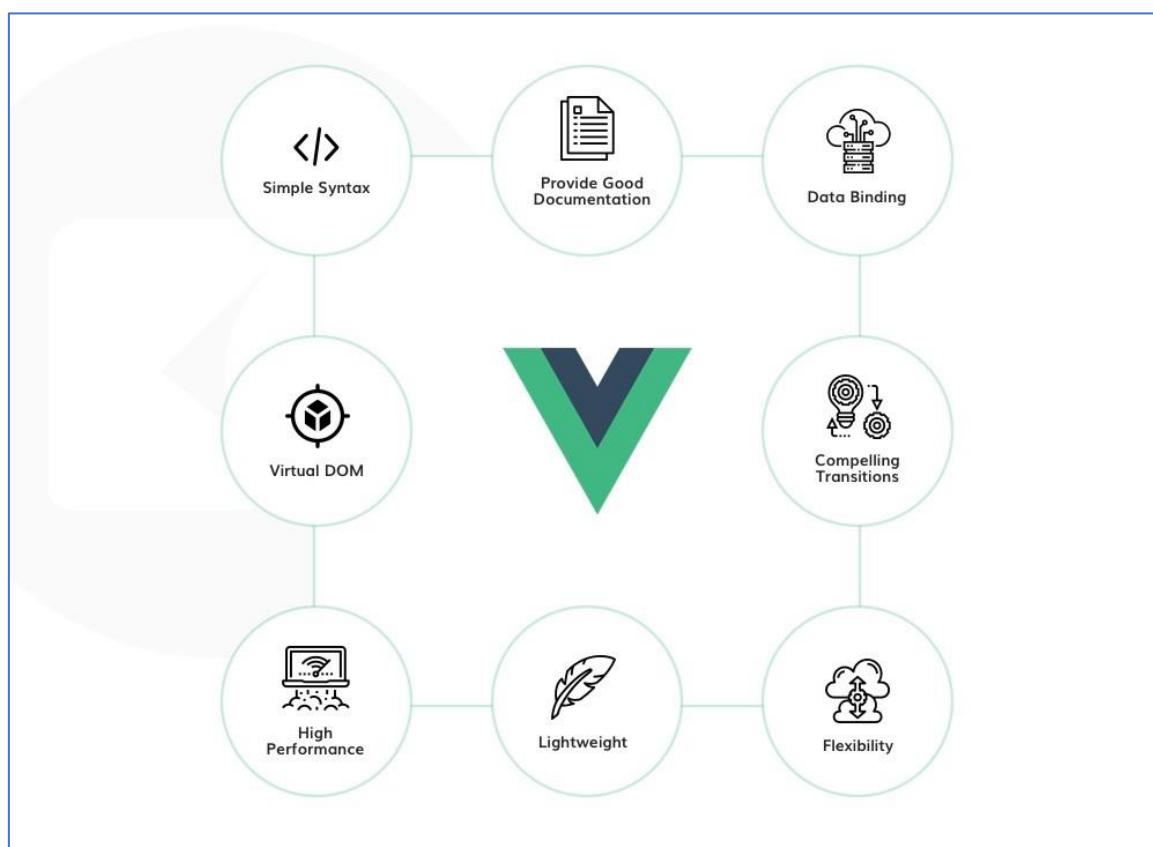


Figure 1. 74 Vue Framework

4) Ember

Ember was chosen the finest Javascript framework in 2015 for its two-way data binding components and other out-of-the-box capabilities. The Ember community is massive, and it's just becoming bigger. On a regular basis, new features and versions were introduced.

This framework is widely used by Google, Microsoft, Heroku, and Netflix. It avoids the need for duplicate and repetitive actions or follows some JS best practices in its fundamental design to maximize the developer's productivity. Ember is primarily concerned with making the life of developers easier.



Figure 1. 75 Ember Framework

5) Bootstrap

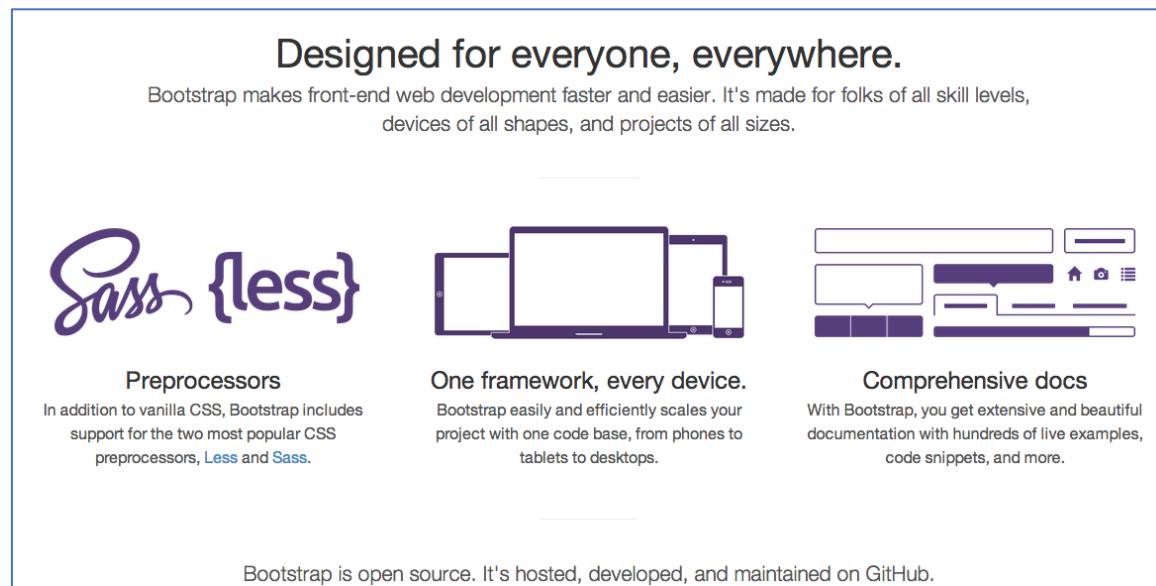
Bootstrap is a web development framework that is both free and open-source. It's intended to make the development of responsive, mobile-first websites easier by providing a set of template design syntax.

To put it another way, Bootstrap makes it easier for web developers to create websites because they don't have to bother about basic commands and functions. It's made up of HTML, CSS, and JS-based scripts for a variety of web design functions and elements.

Bootstrap was created by Mark Otto and Jacob Thornton at Twitter to improve the consistency of tools used on the site while also lowering maintenance. The software was once known as Twitter Blueprint and is also known as Twitter Bootstrap.

Designed for everyone, everywhere.

Bootstrap makes front-end web development faster and easier. It's made for folks of all skill levels, devices of all shapes, and projects of all sizes.



The image shows the official Bootstrap website landing page. At the top, there's a purple header with the Bootstrap logo and navigation links. Below the header, a large banner features the text "Designed for everyone, everywhere." and a subtext about making front-end development faster and easier. The main content area has three columns: "Preprocessors" (with a note about supporting Sass and Less), "One framework, every device." (with an icon showing a laptop, tablet, smartphone, and desktop), and "Comprehensive docs" (with an icon of a document). At the bottom, there's a footer with the text "Bootstrap is open source. It's hosted, developed, and maintained on GitHub."

Sass {less}

Preprocessors
In addition to vanilla CSS, Bootstrap includes support for the two most popular CSS preprocessors, [Less](#) and [Sass](#).

One framework, every device.
Bootstrap easily and efficiently scales your project with one code base, from phones to tablets to desktops.

Comprehensive docs
With Bootstrap, you get extensive and beautiful documentation with hundreds of live examples, code snippets, and more.

Bootstrap is open source. It's hosted, developed, and maintained on GitHub.

Figure 1. 76 Bootstrap Framework

1.15.3 Server-side (back-end) Framework

1) Node.js

Node.js is a runtime that allows us to build server-side applications in JavaScript. Because of JavaScript's popularity, Node.js is widely utilized, and there are many experienced Node.js engineers. It's frequently utilized in real-time online apps. Node.js speeds up and simplifies the development of basic applications and prototypes. It is not, however, suitable for CPU-intensive operations.

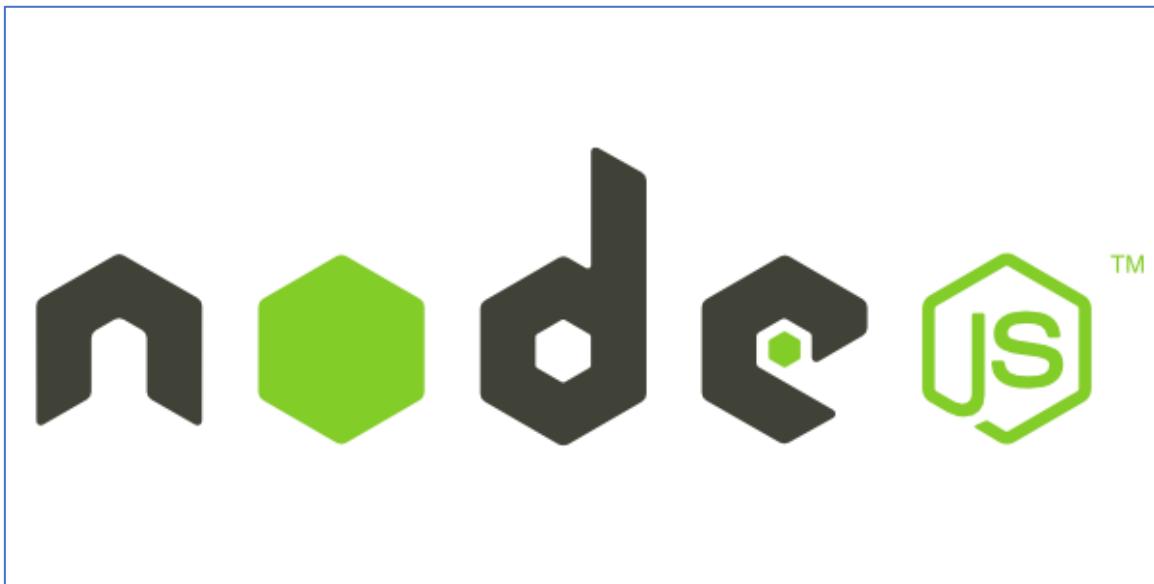


Figure 1. 77 Node.js Framework

2) Express

Express is the most popular backend framework these days, because to the growing popularity of Node.js. It is used by large corporations like as Uber, IBM, and Accenture since it is compatible with other frameworks such as Kraken, Loopback, and Sails. It is a simple, quick, and non-opinionated framework.

Its core functionalities take full advantage of Node.js efficiency without hiding the node's advantages. It supports both full and REST API applications due to its flexibility. There is only one drawback: there is no clear way of accomplishing things, so newcomers may feel overwhelmed.



Figure 1. 78 Express Framework

3) Django

Django is a Python-based model-view-template framework for web development. It is used by website development companies as well as large corporations such as Google, YouTube, and Instagram. Its most prominent functions are authentication and messaging, both of which are integrated in the batteries.

Django emphasizes security and provides techniques and tools for developers to assist in the creation of safe websites and the deployment of security measures in the framework. One example is blocking code execution in the template layer.



Figure 1. 79 Django Framework

4) Rails

is a popular Ruby model-view-controller framework. Airbnb, GitHub, Hulu, and Shopify are all major Rails users. There are numerous useful applications for Rails, which is also considered a beginner-friendly framework.

With library-like dependencies that enhance the functionality of apps, developers may work more quickly and efficiently. There are many tutorials, screencasts, and resources available online if this technology fascinates us to learn it.



Figure 1. 80 Rails Framework

1.16 Tools and techniques chosen to develop the HMS web application

1) Bootstrap

As previously mentioned in Front-End frameworks, Bootstrap is a front-end programming framework for building websites and online apps that is free and open source. The Bootstrap framework uses HTML, CSS, and JavaScript (JS) to make creating responsive, mobile-first websites and apps easier.

Responsive design allows a web page or app to detect the visitor's screen size and orientation and instantly adjust the appearance. The mobile first strategy assumes that smartphones, tablets, and task-specific apps will be used first. Employees' primary tools for getting work done are mobile apps, which address the design requirements of such technologies.

Since my HMS web application is used by both patients and hospital staff, they may use different kinds of devices when logging into the web system. Because of that, to make it easier to view the web system according to the device, I've used bootstrap to match their needs.

And other reason for use Bootstrap is it is simple to grasp. Because of its popularity, there are numerous tutorials and online forums to assist us in getting started. The fact that Bootstrap has a basic file structure is one of the reasons why it is so popular among web developers and designers. Its files are assembled for simple access, and changing them only requires a basic understanding of HTML, CSS, and JS.

Not only that, Bootstrap comes includes a built-in grid system, which reduces the need to create one from scratch. Instead of adding media queries in the CSS file, the grid system uses rows and columns to create a grid within an existing one. Additionally, the grid structure in Bootstrap simplifies the data entry procedure. It has a number of media queries in it, which allows us to build specific breakpoints for each column based on the demands of our web project. In most cases, the default settings are sufficient. After we've created a grid, all I had to do was fill the containers with content.

2) WAMP Server

Windows, Apache, MySQL, and PHP are all abbreviated as WAMP. It's a software stack, which implies that installing WAMP on the operating system installs Apache, MySQL, and PHP (Windows in the case of WAMP). Despite the fact that they can be installed independently, they are frequently bundled together, and for good reason.

It's important to note that WAMP is derived from LAMP (the L stands for Linux). The only difference is that WAMP is used for Windows and LAMP is used for Linux-based operating systems.

Let's take a brief look at what each letter stands for:

W :- There's a "W" for Windows, a LAMP (for Linux), and a MAMP (for Mac).

A :-The letter "A" stands for Apache. The web server software Apache is in charge of serving web pages. When we ask Apache to show us a page, it authorizes our request over HTTP and displays the site.

M :- MySQL is represented by the letter "M." MySQL's role is to function as our server's database management system. It saves all important information, such as our site's content, user profiles, and so on.

P :- PHP is abbreviated as "P". WordPress was written with PHP, which is a computer language. It serves as a bridge for the entire software stack. PHP communicates with MySQL and runs in conjunction with Apache.

We can install and test WordPress on our personal computer instead of on our hosting account (localhost). Since on our machine, WAMP functions as a virtual server I've used this WAMP server to create the HMS web application. Because it is localized on our system and not connected to the internet, it allows us to test all web page functionalities without risk. First and foremost, this eliminates the need to wait for data to be transferred to our site, and it also makes backups easier.

WAMP makes it easier for developers and theme designers to work together. Furthermore, you have the option of playing with our site to our personal liking. However, in order to make the website live, we'll need a hosting service and a domain name. Rather than buying

hosting service for money WAMP serves as a secure environment in which I can work on this HMS website without having to host it online.

And also, another reason is a control panel is also included with WAMP. All of the services stated above (excluding the operating system) will be installed on our local PC once we install the software package. It's a great technique to save time, whether we use WAMP or software packages for other operating systems. We won't have to worry about uploading files to a website, and we'll be able to learn how to build in a secure and worry-free environment.

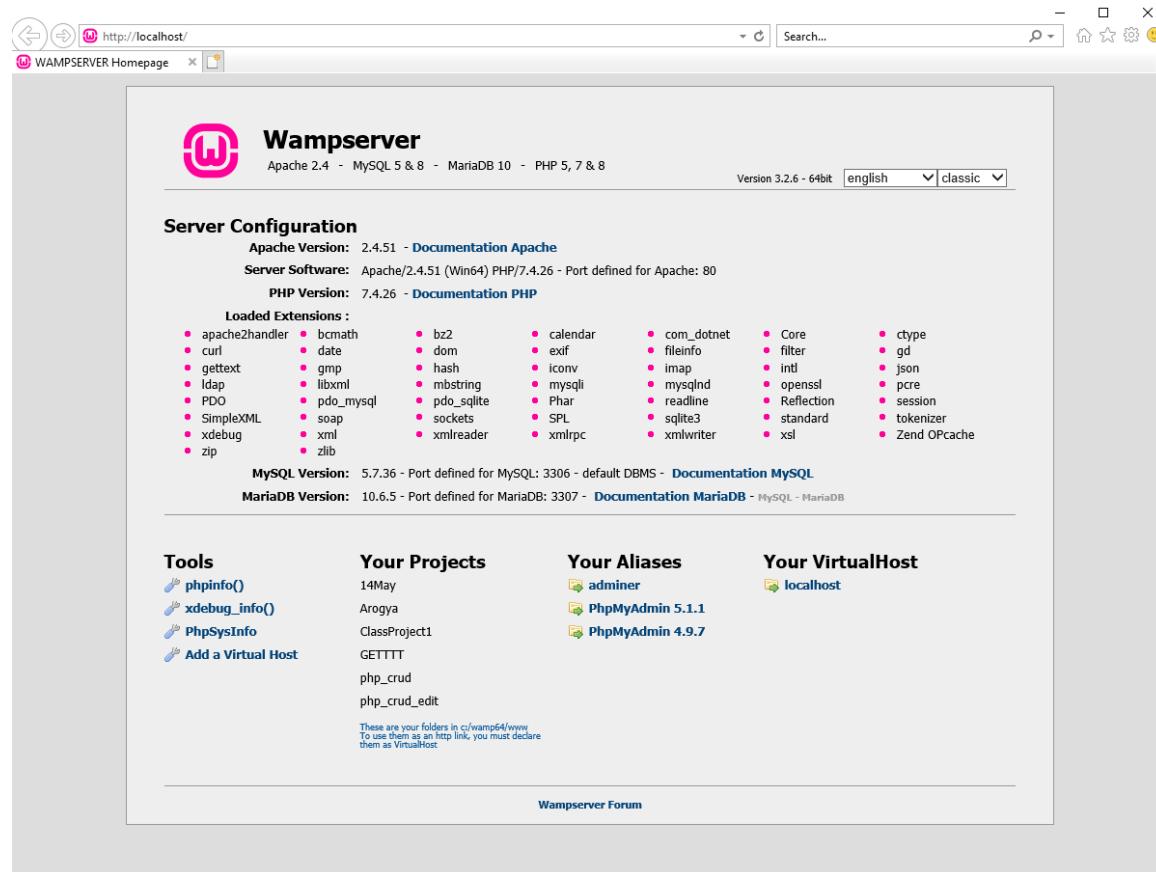


Figure 1. 81 WAMP Server's local host

Task 2

2.1 Relationships between Front-End and Back-End website Technologies

2.1.1 Introduction

Back-end developers provide the infrastructure that allows front-end developers to work on what the user sees. Both are essential components for a well-functioning application or website. When it comes to developing new software, it's not uncommon for businesses to get caught up by the "front-end vs back-end" divide. After all, a significant number of tools are available to assist developers in becoming more "full stack" focused. These terms are both important in web development but are not identical. To increase the website's functionality, each side must communicate and collaborate efficiently as a single unit.



Figure 2. 1 Relationships between Front-End and Back-End website Technologies

2.1.2 What is Front End Development

The front end of a website is the section with which the user directly interacts. It's also referred to as the application's 'client side.' It includes everything that users see and interact with such as text colors and styles, graphics, graphs and tables, buttons, colors, and the navigation menu. The languages used for Front End development are HTML, CSS, and JavaScript. Front End developers handle the structure, design, functionality, and contents of everything shown on browser displays when websites, web applications, or mobile apps are opened. The Front End's primary goals are responsiveness and performance. The developer must guarantee that the site is responsive, meaning that it looks correctly on devices of all sizes. No part of the website should act improperly regardless of screen size.



Figure 2. 2 Front End development

2.1.3 Front-End Programming Languages

The front end is created using some languages, which are described below.

1. HTML :- HTML is an abbreviation for Hypertext Markup Language. It is used to create the front-end of web pages by utilizing a markup language. This is a combination of Hypertext and Markup language. The link between web pages is defined by hypertext. The markup language is used to define written documentation within tags that form the structure of web pages.
2. CSS :- CSS, or "Cascading Style Sheets," is a programming language used to style and layout webpages. It can be used to change the size, spacing, color, and font of content, as well as to add decorative elements such as animations or split content into columns. Styles can be applied to web pages using CSS. More importantly, CSS allows us to do this without relying on the HTML that sets up each web page.
3. JavaScript :- JavaScript is a well-known scripting language that is used to create things on websites in order to make them more interactive for users. It is used to improve the usefulness of a website by allowing it to run exciting games and web-based software.

There are numerous additional languages that can be used for front-end development depending on the framework. For example, Flutter uses Dart, React uses JavaScript, and Django uses Python, and so on.

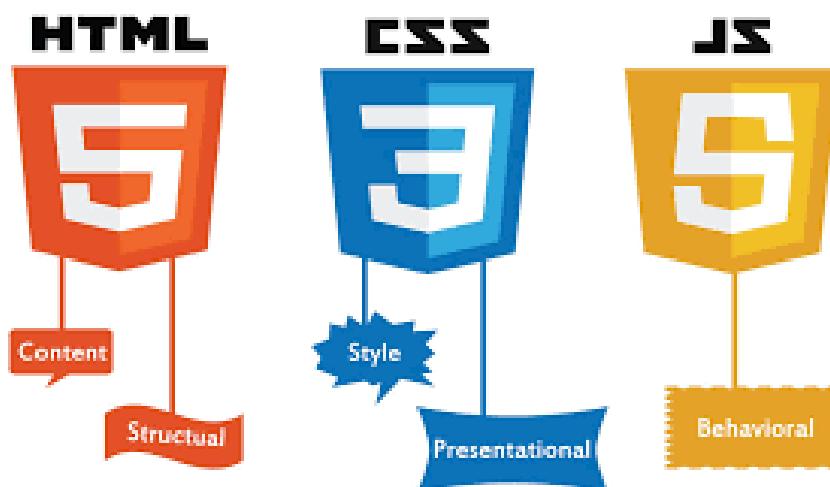


Figure 2. 3 Front End Developing languages

2.1.4 Frameworks and libraries for Front-End Development

1. AngularJs :- AngularJs is an open-source JavaScript front-end framework that is mostly used to create single-page web apps (SPAs). It is a framework that is always growing and extending to provide better ways for constructing web applications. It transforms static HTML into dynamic HTML. It's an open-source project that may be downloaded for free.
2. React.js :- React is a JavaScript library for designing user interfaces that is declarative, efficient, and customizable. ReactJS is an open-source, element front-end library that is only responsible for the application's view layer. React is maintained by Facebook.
3. Bootstrap :- Bootstrap is a collection of free and open-source tools for building responsive websites and online applications. It is the most widely used HTML, CSS, and JavaScript framework for creating mobile-first, responsive websites.
4. jQuery :- Query is a JavaScript library that allows users to write less, achieve more. jQuery's objective is to make it much easier to use JavaScript on our website. jQuery wraps many typical operations that require multiple lines of JavaScript code into methods that can be called with a single line of code.
5. Flutter :- Flutter is a Google-managed open-source UI development project. Dart is the programming language that powers it. Flutter is free and open source, compatible with existing code, and is utilized by developers and organizations all across the world.

2.1.5 What is Back End Development

The term "back-end development" relates to server-side development. It focuses on databases, scripting, and website design. It describes the processes that take place behind the scenes when a user performs any activity on a website. It may be logging into an account or making a purchase from an online business. Back-end developers write code that allows browsers to interface with database information. Backend programming is most commonly seen when reading an article on a blog. This page's frontend is made up of fonts, colors, designs, and so on. While the article's content is rendered from a server and retrieved from a database. This is the application's backend.

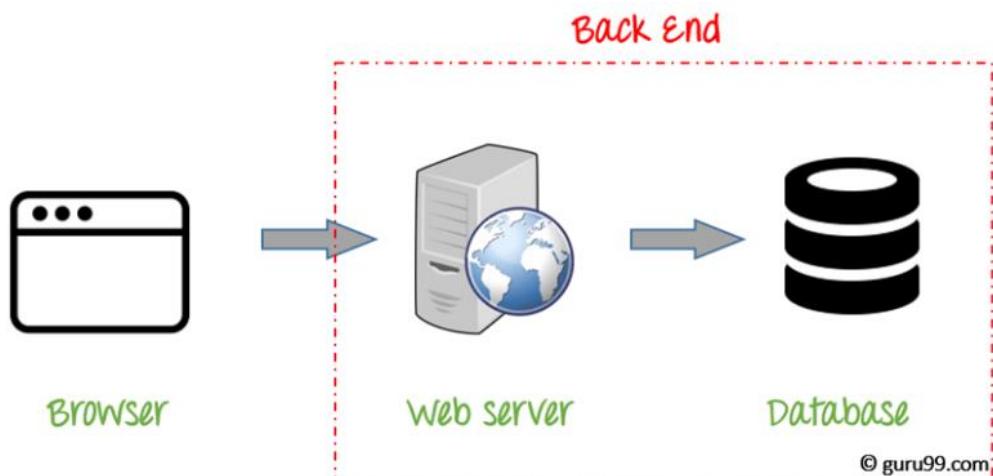


Figure 2. 4 Back End development

2.1.6 Back-End Programming Languages

The back end is created utilizing certain languages, which are described further below.

1. PHP :- PHP is a server-side programming language that was developed primarily for web development. PHP code is referred to as a server-side scripting language since it is executed on the server.
2. C++ :- It is a general-purpose programming language that is frequently used in competitive programming today. It is also used as a backend programming language.
3. Java :- Java is a popular and widely used programming language and platform. It is very scalable. Components for Java are widely available.
4. Python :- Python is a programming language that enables us to operate more quickly and efficiently with systems.
5. JavaScript :- JavaScript is a programming language that may be used on both the front end and the back end.
6. Node.js :- Node.js is a cross-platform open-source runtime environment for executing JavaScript code outside of a browser. It's important to note that NodeJS is not a framework or a programming language. Most people are confused and believe it is a framework or a programming language. We frequently use Node.js to create back-end services such as APIs, Web Apps, and Mobile Apps. Large corporations like as PayPal, Uber, Netflix, and Walmart use it in production.

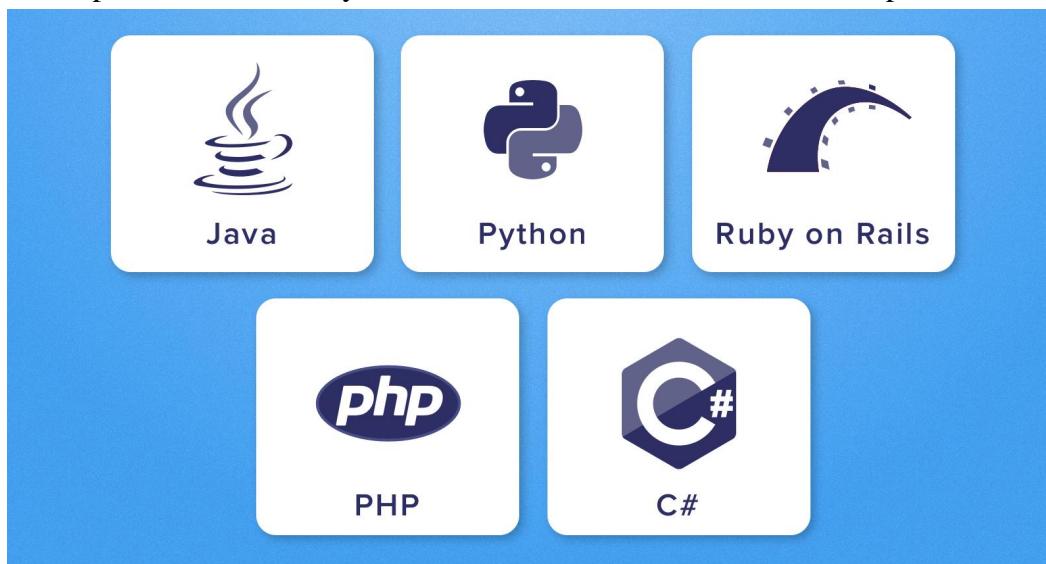


Figure 2. 5 Back-End Programming Languages

2.1.7 Frameworks for Back-End Development

1. Express :- Express is a Node.js backend web application framework that is available as free and open-source software under the MIT License. It is intended for the development of web applications and APIs. This also handles routing, sessions, HTTP requests, error handling, and so on. Express. js's JavaScript library assists programmers in creating efficient and quick web applications.
2. Django :- Django is a Python library that allows users to rapidly and efficiently develop a high-quality Web application that can be used on both the frontend and the backend.
3. Ruby :- Ruby on Rails is a dynamic web application framework that allows developers to build and deploy apps easily and quickly. One of the benefits of Ruby on Rails, which speaks to its popularity among startups, is that it significantly simplifies the software development cycle and allows us to quickly resolve faults and integrate customer business needs in a project, as well as make modifications to them as needed.

2.2 Difference between Frontend and Backend

When speaks about the definition, the frontend of a website is the part that users can see and interact with, such as the graphical user interface (GUI) and the command line, which includes the design, navigating menus, texts, images, videos, and so on. But the backend is the part of the website that users do not view or interact with. It all comes down to how everything works. Both are important in web development, and while they have their differences, they are like two sides of the same coin. The visual aspects of a website that a user may see, and experience are referred to as the frontend. Backend web development, on the other hand, is responsible for everything that happens in the background. It's more of a frontend web experience enabler.

The most popular job title for frontend web development is web designer, and a web designer's purpose is to design and create websites while keeping visual features in mind. Backend developers ensure that the data and systems requested by the frontend application or program are delivered efficiently. Everything that happens in the background is handled by backend developers. Hence, the frontend is sometimes known as the "client-side," as opposed to the backend, which is the "server-side" of the application.

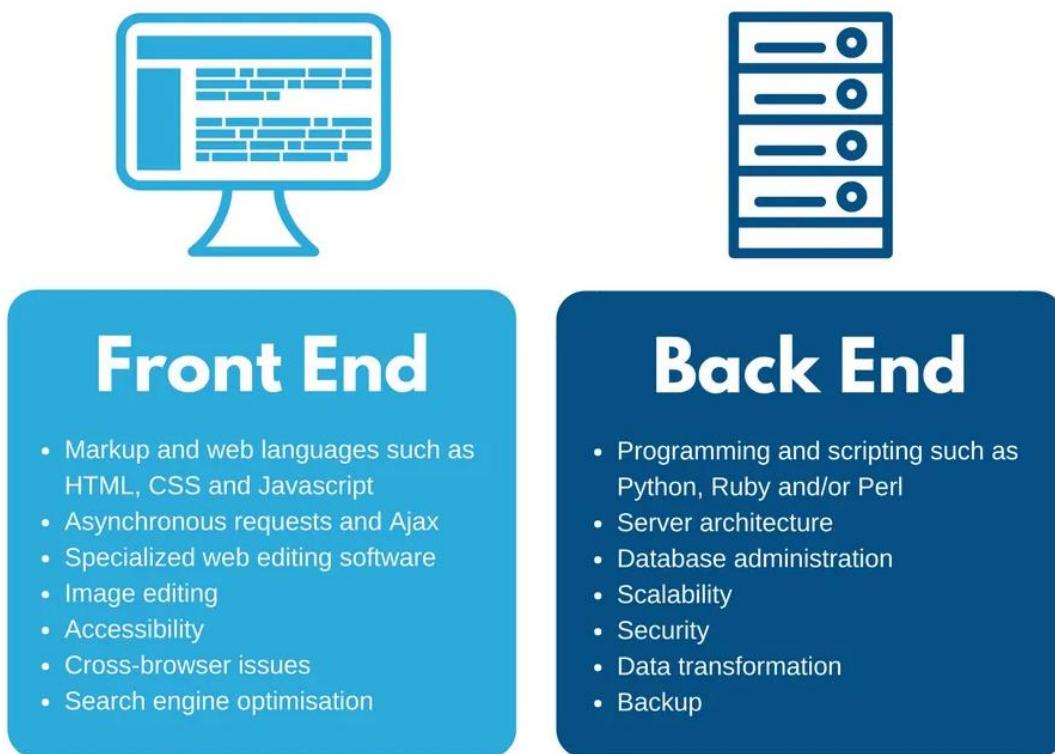


Figure 2. 6 Difference between Frontend and Backend

Table 2. 1 Difference between Frontend and Backend

| Front End | Back End |
|---|--|
| This is the client side of the application. | This is the server side of the application. |
| This is the part where website users can see and interact with. | This constitutes everything that happens behind the scene. |
| This includes everything related to the visual components of the website. | This includes a webserver which communicates with database to serve requests that the frontend presents. |
| Users can touch and experience on their web browsers. | This is the brain of the website which is never visible to users. |
| The essentials of frontend web development are HTML, CSS and JavaScript. | The essentials of backend web development is Ruby, Python and Java, .Net and etc. |

2.3 Full Stack development

Full stack technology refers to the entire depth of a computer system program, and full stack developers work in both the front end and the back end of web development. Everything that a client or site viewer may see and interact with is on the front end. The back end, on the other hand, refers to all of the servers, databases, and other internal architecture that power the program; the end user rarely interacts with this domain directly.

Imagine a restaurant to get a sense of the complete stack. The front end includes the well-decorated, comfortable seating areas where guests can eat. The "back end" consists of the kitchen and pantry, which are normally hidden from the customer's perspective. Chefs (developers) collect permanently stored materials from the pantry (the database), perform operations on them in the kitchen (the server), and then serve fully prepared meals (information) to the user.

Front end developers work to optimize an application's visible sections for web browsers and mobile devices. Front end platforms are often created using HTML, CSS, and JavaScript; however, they can also be created using pre-packaged code libraries or content management systems like as WordPress. Back-end developers, on the other hand, work on improving the software code that communicates with servers, databases, or other proprietary software that sends data to front end interfaces. Full stack developers are those who are well with both front end and back-end development.

In general, full-stack development by a single programmer is not a viable option. However, a custom software development business may employ developers who are adept on all sides, referred to as a full stack developer. They are effective team members because they have the depth of knowledge to understand the larger picture, allowing them to offer methods to optimize the process or remove obstacles that may be limiting the system.



Figure 2. 7 Full stack development

2.4 The three-layer architecture

2.4.1 What is three-layer architecture

A well-known software application architecture is three-layer architecture, which divides applications into three logical and physical computing levels. The presentation layer is made up of the user interface. The application layer, which processes data, and the data layer, which stores and manages the data linked with the application. The main advantage of three-layer architecture is that because each layer runs on its own infrastructure, each layer may be developed simultaneously by a separate development team and updated or scaled as needed without affecting the other layers.

The diagram below shows a common architectural abstraction for Web applications based on the basic three-layer architecture. The presentation layer, which represents the user interface, is the first tier. The application layer executes application logic in the middle tier, whereas the data layer is made up of permanent data storage in the third tier.

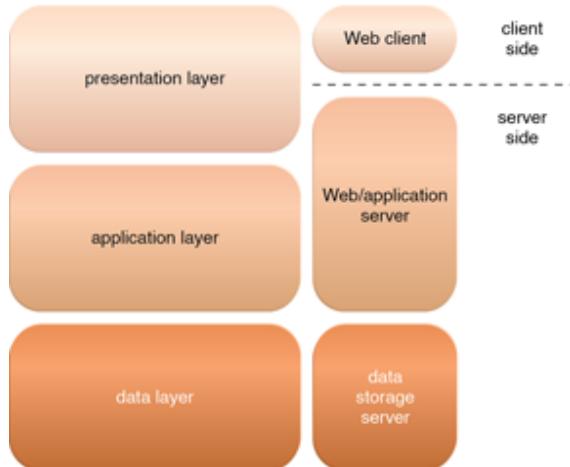


Figure 2. 8 Web applications' three core architectural tiers.

The presentation layer includes client-side and server-side components. Web servers receive client requests and retrieve required resources, both directly as static Web content and indirectly as dynamic Web content generated by application logic. Web servers communicate with application servers to execute the requested application logic, which often includes interaction with one or more underlying databases. All communication in a three-tier application passes through the application tier. The display and data tiers are not able to communicate directly with one another.

2.4.2 Front-End and Back-End web development relationship with 3 layer architecture's Presentation layer, Application layer and Data layer

Presentation Layer →

The presentation layer is the application's user interface and communication layer, where the end user interacts with the application. Its primary function is to present information to and gather data from the user. This top-level tier, for example, can execute in a web browser, as a desktop application, or as a graphical user interface (GUI). Tiers of web presentation are often built with HTML, CSS, and JavaScript. Depending on the platform, desktop apps can be built in a variety of languages. So, front end developers work on Presentation Layer.

Application Layer →

This, often known as the logic tier or middle tier, is the application's core. In this layer, information gathered in presentation tier is processed sometimes in combination with data gathered in the data tier using business logic, or a specific set of business rules. The data tier can also be added, deleted, or modified by the application tier. The application layer is usually written in Python, Java, Perl, PHP, or Ruby and interfaces with the data tier via API calls. So, back-end developers work on Application Layer.

Data Layer →

The database layer, also known as the database layer, data access tier, or back-end, is where the application's data is kept and maintained. This could be a relational database management system like PostgreSQL, MySQL, MariaDB, Oracle, DB2, Informix, or Microsoft SQL Server, or it could be a NoSQL database server like Cassandra, CouchDB, or MongoDB.

2.4.3 Benefits of three-layer architecture

Each layer can run on a different operating system and server platform, such as a web server, application server, or database server, depending on its functional requirements. Furthermore, each layer runs on at least one dedicated server hardware or virtual server, allowing each layer's services to be changed and optimized without affecting the other layers.

- Faster development :- Since each layer can be built simultaneously by multiple teams, a company can bring the application to customers more quickly, and programmers can use the most up-to-date languages and tools for each tier.
- Enhanced scalability :- Each layer can be scaled separately as needed.
- Reduced dependability :- An outage in one layer has a lower chance of affecting the availability or performance of the other layers.
- Enhanced security :- Since the presentation and data layers cannot connect directly, a well-designed application tier can serve as an internal firewall, avoiding SQL injections and other dangerous vulnerabilities.

2.4.5 Three-layer architecture in web development

The layers in web development have different names but serve similar functions.

- The presentation layer is the web server, which serves as the user interface. This is typically a web page or web site, such as an ecommerce site, where the user adds things to the shopping basket, enters payment information, or registers for an account. The content can be static or dynamic, and it is often created with HTML, CSS, and Javascript.
- The application server is the middle layer, containing the business logic used to process user inputs. This layer is frequently written in Python, Ruby, or PHP and runs a framework such as Django, Rails, Symphony, or ASP.NET.
- A database server is a web application's data or backend layer. It is powered by database management software such as MySQL, Oracle, DB2, or PostgreSQL.

2.5 Online Website Creation Tools and Custom-Built Websites

2.5.1 Online Web Creation Tools

Online Website Creation Tools are primarily utilized with website templates. They're becoming increasingly popular, especially among entrepreneurs, bloggers, small enterprises, and people who are short on time. Template websites are pre-made webpages or sets of webpages that are clones of a website's basic core foundation pieces that can then be marginally customized and filled with our own content.

One of the most well-known platforms for using a template design is WordPress. According to estimates from 2017, WordPress was used as the CMS for 50-60% of template-based websites. Template website designs allow businesses to choose from a variety of ready-to-use templates with built-in capabilities. These templates are designed by web professionals and include important elements such as HTML, CSS, and code to ensure that the website runs smoothly. Above all, it facilitates the web building process for customers that need websites fast and inexpensively but don't want anything particularly innovative in terms of design, performance, or functionality.

Pros →

- Templates are ideal for people on a tight budget. Templates will give us all of the tools we need to get the site up and running at a lesser cost.
- We don't need any prior coding experience. Drag and drop features in template web designs allow anybody with any degree of knowledge to construct a website. Regular functions require little to no coding.
- Prebuilt templates also mean lower initial expenses because we won't have to hire someone to put it up if we do it ourselves.
- WE can drastically reduce time it takes to construct the website by choosing a template.

Cons →

- As previously noted, 50-60% of firms used templates for their websites in 2017. With such a high percentage of businesses using the service, we may expect to see a lot of other organizations using the same template as us. In the end, this means that our website will struggle to stand out among the crowd.
- The site's customization options are restricted. The template encourages us to operate inside its constraints, limiting our originality and features. We get exactly what we see.
- The user interface may change depending on the device we use and may not function properly. As a result, there may be inconsistencies in the user experience between devices.
- It's possible that the template we use isn't built to be search engine optimized. Most themes will need some editing to make them SEO-friendly and suitable for our business.
- Some features and technologies may not be appropriate for use on our website. Templates employ a structured framework, which may limit the number of plugins that may be used, limiting our future capability.
- Developers frequently create templates, release them, and then forget about them. This means they aren't maintaining its coding and structure, which can lead to serious security risks if a WordPress platform upgrade is released that breaks the template's compatibility.

2.5.2 Custom-Built Websites

Custom-made websites are designed and built from the ground up (much like a customized tailored suit) specifically for our company, and they typically require a professional and trusted staff to put it all together. It starts with in-depth brainstorming sessions and a well-thought-out strategic plan that reflects the value and brand. Essentially, it's a creative method to figure out who our target market is, how we want to reach out to a broader audience, what our website's purpose is, and how we want to display ourselves on the internet.

Custom-built websites are created through a series of steps that result in a user interface that is unique to the site. Furthermore, the response to user inputs (any action taken by the user) that designers can use to generate a variety of outputs. Above all, custom-built websites will provide us our own "built-from-the-ground-up website" and deliver a lot more unique experience for our users. Custom website designs will provide us more flexibility, but they will take more time to develop and manage.

Pros →

- The website designs created will be one-of-a-kind for firm. It will feature company's branding and will be completely customized to match our current and future growth goals as a corporation in order to get the best possible results.
- The website has been thoroughly tested in a variety of browsers. Because mobile devices accounted for 52.2% of website traffic in 2019, having a consistent look across all browsers and platforms is critical.
- The site will be optimized for search engines (SEO). With Google emphasizing numerous parts of website speed, keywords, meta descriptions, and even URLs being SEO friendly is one of the most important things about our website right now. According to studies, internet consumers prefer URLs that are short, clear, and easy to grasp.

- Will add credibility to company. According to Business 2 Community, customized designs will allow us to establish credibility and establish a professional brand since they will stick out from the crowd.
- Custom-built websites can become a part of our company's ecosystem, allowing us to combine solutions like NetSuite and HubSpot's ERP and CRM systems. These interfaces can help us improve and extend our business by effortlessly synchronizing your business data across all platforms, allowing us to track KPIs, revenues, engagement, performance, and user experience.

Cons →

- A bespoke website will sound expensive because it necessitates greater engagement in the planning, design, and development phases that are specific to our company. We will not, however, be required to pay for additional services such as themes, CMS services, or subscriptions.
- A custom-built website will take longer to develop than a template one. This is due to the fact that there will be a back-and-forth procedure between the consumer and the custom website supplier. Furthermore, writing the lines of code and checking that they are working properly takes more time.
- Our site will expand in parallel with your company, necessitating the hiring of a design and development team to add new pages and features.

Table 2. 2 Differences between Online Website Creation Tools and Custom-Built Websites

| | Online Web creation tools | Custom Built websites |
|--------------------|--|--|
| Flexibility | <ul style="list-style-type: none"> * Ideal for people on a tight budget. * Don't need any coding experience. * Very easy to create a website. * Finding the best matching online tool or template is very hard. * May limit the number of plugins that may be used, limiting our future capability. * Customization options are restricted. The template encourages us to operate inside its constraints, limiting our originality and features. | <ul style="list-style-type: none"> * Very expensive. * Need coding experience. * Very hard to create website. * Can create websites according to customer needs without worrying about finding templates or tools. * No limitations. * No limitations. |
| Performance | <ul style="list-style-type: none"> * Takes less time to make a website. | <ul style="list-style-type: none"> * Takes more time to make a website. |

| | | |
|-----------------------------|--|---|
| | <ul style="list-style-type: none"> * Very easy to create a website. * Most of time the site is hard be optimized according to SEO tactics. * Can lead to security risks. * Some tools aren't especially well-made or maintained. | <ul style="list-style-type: none"> * Very hard to create website. * The site can be optimized according to SEO tactics. * Can handle security risks. * For custom web building doesn't need or matter any sophisticated or maintained tools and techniques. |
| Functionality | <ul style="list-style-type: none"> * Come with a vast number of built-in features. * Comes with a wide range of powerful features. * Hard to synchronize business data in all platforms according to customer needs. | <ul style="list-style-type: none"> * User has to create functionalities from scratch. * User must be fluent in web developing to create powerful features. * We can synchronize business data across all platforms according to our needs easily. |
| User Experience (UX) | <ul style="list-style-type: none"> * Have to use already made template or tools. * Changing and customizing user experience according to customer need is hard. | <ul style="list-style-type: none"> * Can give unique experience for users. * Can customize user experience according to match client's current and future growth goal. |

| | | |
|----------------------------|---|---|
| | <ul style="list-style-type: none"> * There may be inconsistencies in the user experience between devices if user interface change depending on the device. * Limited in navigation capability. | <ul style="list-style-type: none"> * There is no such issues. If there is, we can handle it properly. * No limitation. |
| User Interface (UI) | <ul style="list-style-type: none"> * Have to use already made template or tools. * Changing and customizing user interface according to customer need would be hard. * Generally, they come with a vast number of widgets and settings to make the setup process simple for anyone of any skill level. * Need to modify or update the given design or tool. * The user interface may change depending on the device we use and may not function properly. * Limited in graphics design. | <ul style="list-style-type: none"> * Can give unique interface for users. * Can customize user interface according to match client's current and future growth goal. * User has to create interface from scratch. * Need to go through planning, design, and development phases. * There are no such issues. If there is, we can handle it properly. * No limitation. |

2.6 Tools and techniques used to design the Hospital Management System.

There are many websites offering website templates and tools to create Hospital Management System. Some of them are as following.

- <https://themeforest.net/>
- <https://dribbble.com/>
- <https://www.smarteyeapps.com/>
- <https://www.templatemonster.com/>
- <https://colorlib.com/>

Every day, the number of web development tools grows. Finding the proper software to complete the job might be difficult at times. To get started, here's a list of essential frontend development tools.

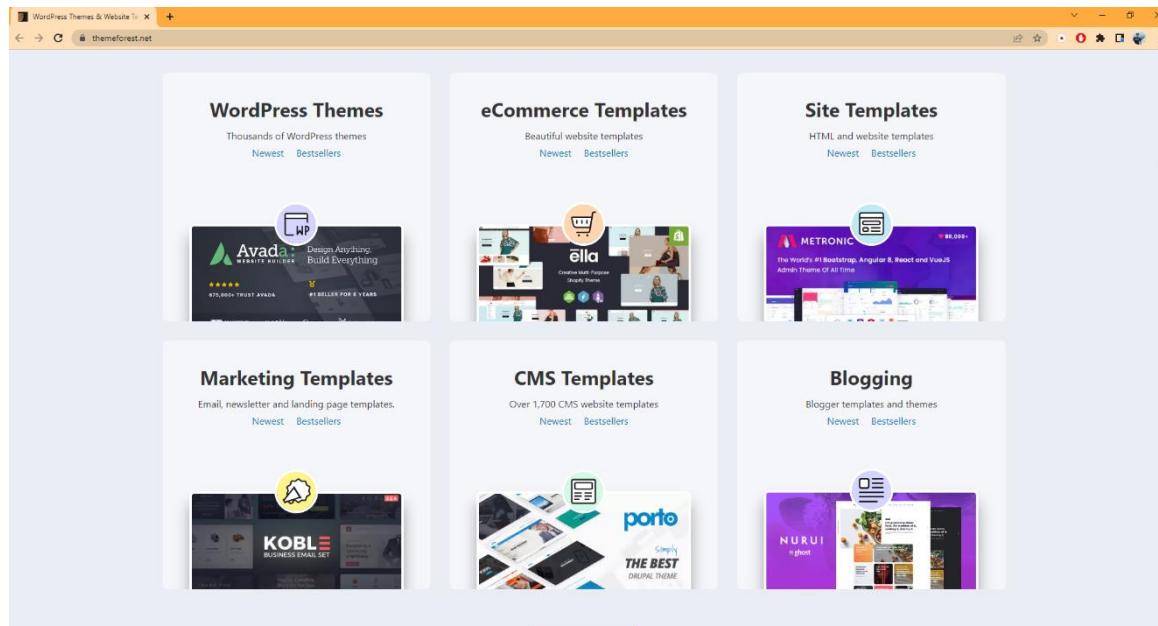


Figure 2. 9 Themeforest.net home page

1) Sublime Text

Sublime Text is a top-notch code editor with a well-designed, super-efficient, and lightning-fast user interface. There are a few that perform this well, but Sublime Text is perhaps the best (and most popular).

The secret to Sublime's success is the program's wide array of keyboard shortcuts, such as the ability to execute simultaneous editing (making the same interactive changes to several chosen areas) and speedy navigation to files, symbols, and lines, which are all managed by a one-man development team. And when we're spending 8+ hours a day with our editor, those few seconds saved for each operation add up quickly. Because of that, I've used Sublime Text when code editing.

```

C:\wamp\www\Arogya\Page_Files\Medicines.php - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
OPEN FILES <-- Staff.php <-- Medicines.php
1 <?php
2
3 session_start();
4 if(isset($_SESSION['User'])){
5 header("location:../INDEX.php");
6 }
7
8 >
9
10 <!-- Header -->
11 <?php include "../header.php";>
12
13 <!-- Nav Bar -->
14 <nav class="navbar navbar-light bg-info navbar-expand-sm fixed-top">
15
16 <div class="container">
17 <a href="#" class="navbar-brand" > AROGYA </a>
18
19 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavDropdown"> <span class="navbar-toggler-icon"></span> </button>
20
21 <div class="collapse navbar-collapse" id="navbarNavDropdown">
22 <!-- Nav Bar can collapse like a drop down button -->
23
24 <ul class="navbar-nav ms-auto">
25 <li class="nav-item" > <a href="Details_Pages/Home.php" class="nav-link" > Home </a> </li>
26 <li class="nav-item" > <a href="Details_Pages/About.php" class="nav-link" > About Us </a> </li>
27 <li class="nav-item" > <a href="Details_Pages/Blog.php" class="nav-link" > Blog </a> </li>
28 <li class="nav-item" > <a href="Details_Pages/Contact.php" class="nav-link" > Contact Us </a> </li>
29
30 <li class="nav-item dropdown">
31 <a href="#" class="nav-link dropdown-toggle" href="#" id="navbarDropdownNavDropdown" data-toggle="dropdown" data-link="navbarDropdownNavDropdown" data-menu="navbarDropdownMenuLink" data-menu-link="navbarDropdownNavDropdownLink" data-target="#"> Services </a>
32
33 <div class="dropdown-menu">
34 <a class="dropdown-item" href="Staff.php">Staff</a>
35 <a class="dropdown-item" href="Patients.php">Patients</a>
36 <a class="dropdown-item" href="Wards.php">Wards</a>
37 <a class="dropdown-item" href="Rooms.php">Rooms</a>
38 </div>
39 </li>
40
41 </ul>
42
43 </div>
44
45 <div class="container">
46 <nav class="navbar navbar-light bg-info navbar-expand-sm fixed-top">
47
48 <div class="container">
49 <a href="#" class="navbar-brand" > AROGYA </a>
50
51 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavDropdown" data-link="navbarDropdownNavDropdown" data-menu="navbarDropdownMenuLink" data-menu-link="navbarDropdownNavDropdownLink" data-target="#"> Services </button>
52
53 <div class="collapse navbar-collapse" id="navbarNavDropdown" >
54 <!-- Nav Bar can collapse like a drop down button -->
55
56 <ul class="navbar-nav ms-auto">
57 <li class="nav-item" > <a href="Details_Pages/Home.php" class="nav-link" > Home </a> </li>
58 <li class="nav-item" > <a href="Details_Pages/About.php" class="nav-link" > About Us </a> </li>
59 <li class="nav-item" > <a href="Details_Pages/Blog.php" class="nav-link" > Blog </a> </li>
60 <li class="nav-item" > <a href="Details_Pages/Contact.php" class="nav-link" > Contact Us </a> </li>
61
62 <li class="nav-item dropdown">
63 <a href="#" class="nav-link dropdown-toggle" href="#" id="navbarDropdownNavDropdown" data-link="#"> Services </a>
64
65 <div class="dropdown-menu">
66 <a class="dropdown-item" href="Staff.php">Staff</a>
67 <a class="dropdown-item" href="Patients.php">Patients</a>
68 <a class="dropdown-item" href="Wards.php">Wards</a>
69 <a class="dropdown-item" href="Rooms.php">Rooms</a>
70
71 </div>
72 </li>
73
74 </ul>
75 </div>
76 </div>
77
78 </div>
79
80 </div>
81
82 </div>
83
84 </div>
85
86 </div>
87
88 </div>
89
90 </div>
91
92 </div>
93
94 </div>
95
96 </div>
97
98 </div>
99
100 </div>
101
102 </div>
103
104 </div>
105
106 </div>
107
108 </div>
109
110 </div>
111
112 </div>
113
114 </div>
115
116 </div>
117
118 </div>
119
120 </div>
121
122 </div>
123
124 </div>
125
126 </div>
127
128 </div>
129
130 </div>
131
132 </div>
133
134 </div>
135
136 </div>
137
138 </div>
139
140 </div>
141
142 </div>
143
144 </div>
145
146 </div>
147
148 </div>
149
150 </div>
151
152 </div>
153
154 </div>
155
156 </div>
157
158 </div>
159
160 </div>
161
162 </div>
163
164 </div>
165
166 </div>
167
168 </div>
169
170 </div>
171
172 </div>
173
174 </div>
175
176 </div>
177
178 </div>
179
180 </div>
181
182 </div>
183
184 </div>
185
186 </div>
187
188 </div>
189
190 </div>
191
192 </div>
193
194 </div>
195
196 </div>
197
198 </div>
199
200 </div>
201
202 </div>
203
204 </div>
205
206 </div>
207
208 </div>
209
210 </div>
211
212 </div>
213
214 </div>
215
216 </div>
217
218 </div>
219
220 </div>
221
222 </div>
223
224 </div>
225
226 </div>
227
228 </div>
229
230 </div>
231
232 </div>
233
234 </div>
235
236 </div>
237
238 </div>
239
240 </div>
241
242 </div>
243
244 </div>
245
246 </div>
247
248 </div>
249
250 </div>
251
252 </div>
253
254 </div>
255
256 </div>
257
258 </div>
259
260 </div>
261
262 </div>
263
264 </div>
265
266 </div>
267
268 </div>
269
270 </div>
271
272 </div>
273
274 </div>
275
276 </div>
277
278 </div>
279
280 </div>
281
282 </div>
283
284 </div>
285
286 </div>
287
288 </div>
289
290 </div>
291
292 </div>
293
294 </div>
295
296 </div>
297
298 </div>
299
300 </div>
301
302 </div>
303
304 </div>
305
306 </div>
307
308 </div>
309
310 </div>
311
312 </div>
313
314 </div>
315
316 </div>
317
318 </div>
319
320 </div>
321
322 </div>
323
324 </div>
325
326 </div>
327
328 </div>
329
330 </div>
331
332 </div>
333
334 </div>
335
336 </div>
337
338 </div>
339
340 </div>
341
342 </div>
343
344 </div>
345
346 </div>
347
348 </div>
349
350 </div>
351
352 </div>
353
354 </div>
355
356 </div>
357
358 </div>
359
360 </div>
361
362 </div>
363
364 </div>
365
366 </div>
367
368 </div>
369
370 </div>
371
372 </div>
373
374 </div>
375
376 </div>
377
378 </div>
379
380 </div>
381
382 </div>
383
384 </div>
385
386 </div>
387
388 </div>
389
390 </div>
391
392 </div>
393
394 </div>
395
396 </div>
397
398 </div>
399
400 </div>
401
402 </div>
403
404 </div>
405
406 </div>
407
408 </div>
409
410 </div>
411
412 </div>
413
414 </div>
415
416 </div>
417
418 </div>
419
420 </div>
421
422 </div>
423
424 </div>
425
426 </div>
427
428 </div>
429
430 </div>
431
432 </div>
433
434 </div>
435
436 </div>
437
438 </div>
439
440 </div>
441
442 </div>
443
444 </div>
445
446 </div>
447
448 </div>
449
450 </div>
451
452 </div>
453
454 </div>
455
456 </div>
457
458 </div>
459
460 </div>
461
462 </div>
463
464 </div>
465
466 </div>
467
468 </div>
469
470 </div>
471
472 </div>
473
474 </div>
475
476 </div>
477
478 </div>
479
480 </div>
481
482 </div>
483
484 </div>
485
486 </div>
487
488 </div>
489
490 </div>
491
492 </div>
493
494 </div>
495
496 </div>
497
498 </div>
499
500 </div>
501
502 </div>
503
504 </div>
505
506 </div>
507
508 </div>
509
510 </div>
511
512 </div>
513
514 </div>
515
516 </div>
517
518 </div>
519
520 </div>
521
522 </div>
523
524 </div>
525
526 </div>
527
528 </div>
529
530 </div>
531
532 </div>
533
534 </div>
535
536 </div>
537
538 </div>
539
540 </div>
541
542 </div>
543
544 </div>
545
546 </div>
547
548 </div>
549
550 </div>
551
552 </div>
553
554 </div>
555
556 </div>
557
558 </div>
559
560 </div>
561
562 </div>
563
564 </div>
565
566 </div>
567
568 </div>
569
570 </div>
571
572 </div>
573
574 </div>
575
576 </div>
577
578 </div>
579
580 </div>
581
582 </div>
583
584 </div>
585
586 </div>
587
588 </div>
589
590 </div>
591
592 </div>
593
594 </div>
595
596 </div>
597
598 </div>
599
600 </div>
601
602 </div>
603
604 </div>
605
606 </div>
607
608 </div>
609
610 </div>
611
612 </div>
613
614 </div>
615
616 </div>
617
618 </div>
619
620 </div>
621
622 </div>
623
624 </div>
625
626 </div>
627
628 </div>
629
630 </div>
631
632 </div>
633
634 </div>
635
636 </div>
637
638 </div>
639
640 </div>
641
642 </div>
643
644 </div>
645
646 </div>
647
648 </div>
649
650 </div>
651
652 </div>
653
654 </div>
655
656 </div>
657
658 </div>
659
660 </div>
661
662 </div>
663
664 </div>
665
666 </div>
667
668 </div>
669
670 </div>
671
672 </div>
673
674 </div>
675
676 </div>
677
678 </div>
679
680 </div>
681
682 </div>
683
684 </div>
685
686 </div>
687
688 </div>
689
690 </div>
691
692 </div>
693
694 </div>
695
696 </div>
697
698 </div>
699
700 </div>
701
702 </div>
703
704 </div>
705
706 </div>
707
708 </div>
709
710 </div>
711
712 </div>
713
714 </div>
715
716 </div>
717
718 </div>
719
720 </div>
721
722 </div>
723
724 </div>
725
726 </div>
727
728 </div>
729
730 </div>
731
732 </div>
733
734 </div>
735
736 </div>
737
738 </div>
739
740 </div>
741
742 </div>
743
744 </div>
745
746 </div>
747
748 </div>
749
750 </div>
751
752 </div>
753
754 </div>
755
756 </div>
757
758 </div>
759
760 </div>
761
762 </div>
763
764 </div>
765
766 </div>
767
768 </div>
769
770 </div>
771
772 </div>
773
774 </div>
775
776 </div>
777
778 </div>
779
780 </div>
781
782 </div>
783
784 </div>
785
786 </div>
787
788 </div>
789
790 </div>
791
792 </div>
793
794 </div>
795
796 </div>
797
798 </div>
799
800 </div>
801
802 </div>
803
804 </div>
805
806 </div>
807
808 </div>
809
810 </div>
811
812 </div>
813
814 </div>
815
816 </div>
817
818 </div>
819
820 </div>
821
822 </div>
823
824 </div>
825
826 </div>
827
828 </div>
829
830 </div>
831
832 </div>
833
834 </div>
835
836 </div>
837
838 </div>
839
840 </div>
841
842 </div>
843
844 </div>
845
846 </div>
847
848 </div>
849
850 </div>
851
852 </div>
853
854 </div>
855
856 </div>
857
858 </div>
859
860 </div>
861
862 </div>
863
864 </div>
865
866 </div>
867
868 </div>
869
870 </div>
871
872 </div>
873
874 </div>
875
876 </div>
877
878 </div>
879
880 </div>
881
882 </div>
883
884 </div>
885
886 </div>
887
888 </div>
889
890 </div>
891
892 </div>
893
894 </div>
895
896 </div>
897
898 </div>
899
900 </div>
901
902 </div>
903
904 </div>
905
906 </div>
907
908 </div>
909
910 </div>
911
912 </div>
913
914 </div>
915
916 </div>
917
918 </div>
919
920 </div>
921
922 </div>
923
924 </div>
925
926 </div>
927
928 </div>
929
930 </div>
931
932 </div>
933
934 </div>
935
936 </div>
937
938 </div>
939
940 </div>
941
942 </div>
943
944 </div>
945
946 </div>
947
948 </div>
949
950 </div>
951
952 </div>
953
954 </div>
955
956 </div>
957
958 </div>
959
960 </div>
961
962 </div>
963
964 </div>
965
966 </div>
967
968 </div>
969
970 </div>
971
972 </div>
973
974 </div>
975
976 </div>
977
978 </div>
979
980 </div>
981
982 </div>
983
984 </div>
985
986 </div>
987
988 </div>
989
990 </div>
991
992 </div>
993
994 </div>
995
996 </div>
997
998 </div>
999
1000 </div>
1001
1002 </div>
1003
1004 </div>
1005
1006 </div>
1007
1008 </div>
1009
1010 </div>
1011
1012 </div>
1013
1014 </div>
1015
1016 </div>
1017
1018 </div>
1019
1020 </div>
1021
1022 </div>
1023
1024 </div>
1025
1026 </div>
1027
1028 </div>
1029
1030 </div>
1031
1032 </div>
1033
1034 </div>
1035
1036 </div>
1037
1038 </div>
1039
1040 </div>
1041
1042 </div>
1043
1044 </div>
1045
1046 </div>
1047
1048 </div>
1049
1050 </div>
1051
1052 </div>
1053
1054 </div>
1055
1056 </div>
1057
1058 </div>
1059
1060 </div>
1061
1062 </div>
1063
1064 </div>
1065
1066 </div>
1067
1068 </div>
1069
1070 </div>
1071
1072 </div>
1073
1074 </div>
1075
1076 </div>
1077
1078 </div>
1079
1080 </div>
1081
1082 </div>
1083
1084 </div>
1085
1086 </div>
1087
1088 </div>
1089
1090 </div>
1091
1092 </div>
1093
1094 </div>
1095
1096 </div>
1097
1098 </div>
1099
1100 </div>
1101
1102 </div>
1103
1104 </div>
1105
1106 </div>
1107
1108 </div>
1109
1110 </div>
1111
1112 </div>
1113
1114 </div>
1115
1116 </div>
1117
1118 </div>
1119
1120 </div>
1121
1122 </div>
1123
1124 </div>
1125
1126 </div>
1127
1128 </div>
1129
1130 </div>
1131
1132 </div>
1133
1134 </div>
1135
1136 </div>
1137
1138 </div>
1139
1140 </div>
1141
1142 </div>
1143
1144 </div>
1145
1146 </div>
1147
1148 </div>
1149
1150 </div>
1151
1152 </div>
1153
1154 </div>
1155
1156 </div>
1157
1158 </div>
1159
1160 </div>
1161
1162 </div>
1163
1164 </div>
1165
1166 </div>
1167
1168 </div>
1169
1170 </div>
1171
1172 </div>
1173
1174 </div>
1175
1176 </div>
1177
1178 </div>
1179
1180 </div>
1181
1182 </div>
1183
1184 </div>
1185
1186 </div>
1187
1188 </div>
1189
1190 </div>
1191
1192 </div>
1193
1194 </div>
1195
1196 </div>
1197
1198 </div>
1199
1200 </div>
1201
1202 </div>
1203
1204 </div>
1205
1206 </div>
1207
1208 </div>
1209
1210 </div>
1211
1212 </div>
1213
1214 </div>
1215
1216 </div>
1217
1218 </div>
1219
1220 </div>
1221
1222 </div>
1223
1224 </div>
1225
1226 </div>
1227
1228 </div>
1229
1230 </div>
1231
1232 </div>
1233
1234 </div>
1235
1236 </div>
1237
1238 </div>
1239
1240 </div>
1241
1242 </div>
1243
1244 </div>
1245
1246 </div>
1247
1248 </div>
1249
1250 </div>
1251
1252 </div>
1253
1254 </div>
1255
1256 </div>
1257
1258 </div>
1259
1260 </div>
1261
1262 </div>
1263
1264 </div>
1265
1266 </div>
1267
1268 </div>
1269
1270 </div>
1271
1272 </div>
1273
1274 </div>
1275
1276 </div>
1277
1278 </div>
1279
1280 </div>
1281
1282 </div>
1283
1284 </div>
1285
1286 </div>
1287
1288 </div>
1289
1290 </div>
1291
1292 </div>
1293
1294 </div>
1295
1296 </div>
1297
1298 </div>
1299
1300 </div>
1301
1302 </div>
1303
1304 </div>
1305
1306 </div>
1307
1308 </div>
1309
1310 </div>
1311
1312 </div>
1313
1314 </div>
1315
1316 </div>
1317
1318 </div>
1319
1320 </div>
1321
1322 </div>
1323
1324 </div>
1325
1326 </div>
1327
1328 </div>
1329
1330 </div>
1331
1332 </div>
1333
1334 </div>
1335
1336 </div>
1337
1338 </div>
1339
1340 </div>
1341
1342 </div>
1343
1344 </div>
1345
1346 </div>
1347
1348 </div>
1349
1350 </div>
1351
1352 </div>
1353
1354 </div>
1355
1356 </div>
1357
1358 </div>
1359
1360 </div>
1361
1362 </div>
1363
1364 </div>
1365
1366 </div>
1367
1368 </div>
1369
1370 </div>
1371
1372 </div>
1373
1374 </div>
1375
1376 </div>
1377
1378 </div>
1379
1380 </div>
1381
1382 </div>
1383
1384 </div>
1385
1386 </div>
1387
1388 </div>
1389
1390 </div>
1391
1392 </div>
1393
1394 </div>
1395
1396 </div>
1397
1398 </div>
1399
1400 </div>
1401
1402 </div>
1403
1404 </div>
1405
1406 </div>
1407
1408 </div>
1409
1410 </div>
1411
1412 </div>
1413
1414 </div>
1415
1416 </div>
1417
1418 </div>
1419
1420 </div>
1421
1422 </div>
1423
1424 </div>
1425
1426 </div>
1427
1428 </div>
1429
1430 </div>
1431
1432 </div>
1433
1434 </div>
1435
1436 </div>
1437
1438 </div>
1439
1440 </div>
1441
1442 </div>
1443
1444 </div>
1445
1446 </div>
1447
1448 </div>
1449
1450 </div>
1451
1452 </div>
1453
1454 </div>
1455
1456 </div>
1457
1458 </div>
1459
1460 </div>
1461
1462 </div>
1463
1464 </
```

2) Chrome Developer Tools

We can change our HTML and CSS in real time, or debug our JavaScript, all while watching a detailed performance analysis of our website, thanks to Google's built-in Chrome Developer Tools. They come pre-installed in Chrome and Safari and give developers access to the inner workings of a web application. A palette of network tools can also help in the optimization of the loading routines, while a timeline provides a better knowledge of what the browser is doing at any given time. Because of that, I've used Chrome Developer Tools when code editing for the Hospital Management System.

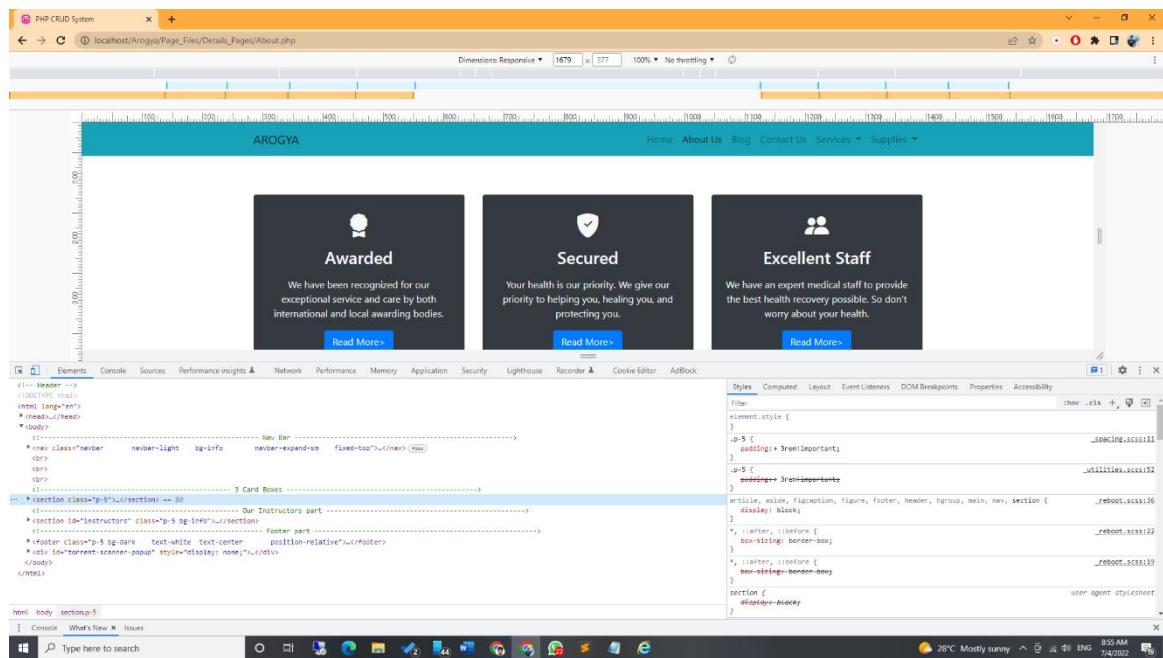
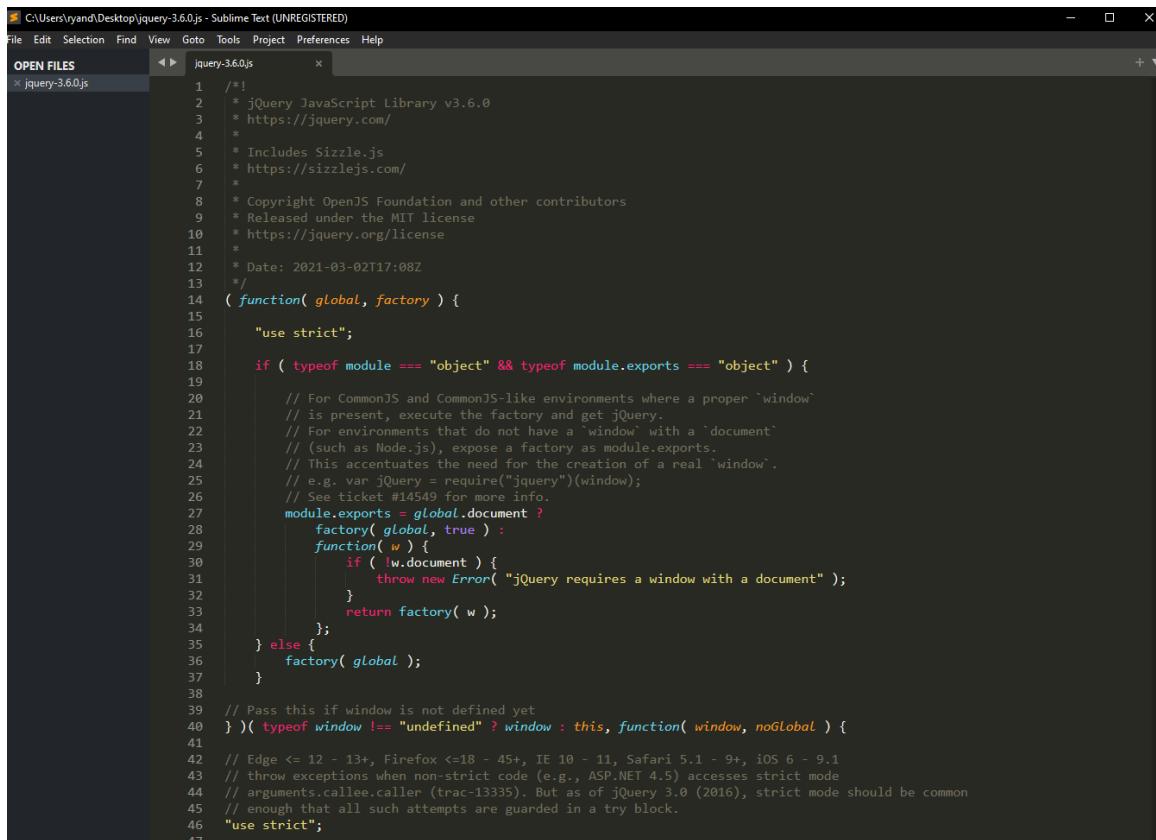


Figure 2. 11 Using Chrome Developer Tools for developing

3) jQuery

Developers have long considered JavaScript to be an essential frontend language, despite its flaws: it's littered with browser bugs, and its rather complicated and unapproachable syntax meant that functionality was frequently compromised. That is, until 2006, when jQuery, a quick, lightweight, cross-platform JavaScript library targeted at easing the frontend process, debuted. jQuery gave developers more freedom to create animations, add plug-ins, and even navigate documents by abstracting a lot of the functionality that was previously left to them to figure out on their own. Because of that, I've used jQuery tools also when code editing for the Hospital Management System.

A screenshot of the Sublime Text editor showing the contents of the jquery-3.6.0.js file. The file is an open-source JavaScript library for the web. The code is well-structured with comments explaining the logic. It handles different environments like CommonJS and browser environments, ensuring compatibility. The code includes checks for 'window' and 'document' objects, and handles strict mode. The file is 47 lines long.

```
/*!
 *  jQuery JavaScript Library v3.6.0
 *  https://jquery.com/
 *
 *  Includes Sizzle.js
 *  https://sizzlejs.com/
 *
 *  Copyright OpenJS Foundation and other contributors
 *  Released under the MIT license
 *  https://jquery.org/license
 *
 *  Date: 2021-03-02T17:08Z
 */
(function( global, factory ) {
    "use strict";
    if ( typeof module === "object" && typeof module.exports === "object" ) {
        // For CommonJS and CommonJS-like environments where a proper `window`
        // is present, execute the factory and get jQuery.
        // For environments that do not have a `window` with a `document`
        // (such as Node.js), expose a factory as module.exports.
        // This accentuates the need for the creation of a real `window`.
        // e.g. var jQuery = require("jquery")(window);
        // See ticket #14549 for more info.
        module.exports = global.document ?
            factory( global, true ) :
            function( w ) {
                if ( !w.document ) {
                    throw new Error( "jQuery requires a window with a document" );
                }
                return factory( w );
            };
    } else {
        factory( global );
    }
}
// Pass this if window is not defined yet
} )( typeof window !== "undefined" ? window : this, function( window, noGlobal ) {
    // Edge <= 12 - 13+, Firefox <=18 - 45+, IE 10 - 11, Safari 5.1 - 9+, iOS 6 - 9.1
    // throw exceptions when non-strict code (e.g., ASP.NET 4.5) accesses strict mode
    // arguments.callee.caller (trac-13335). But as of jQuery 3.0 (2016), strict mode should be common
    // enough that all such attempts are guarded in a try block.
    "use strict";
})
```

Figure 2. 12 Using jquery-3.6.0.js for development

4) CodePen

Despite the fact that it has been around since 2012, the growing number of individuals learning programming means that 2022 will be another big year for the frontend community's favorite tool. There's almost no better way to show off the HTML, CSS, and JavaScript snippets than embedding them, which is why they're becoming more frequent in online coding tools. Because of that, I've used CodePen tools also for some parts when code editing for the Hospital Management System.

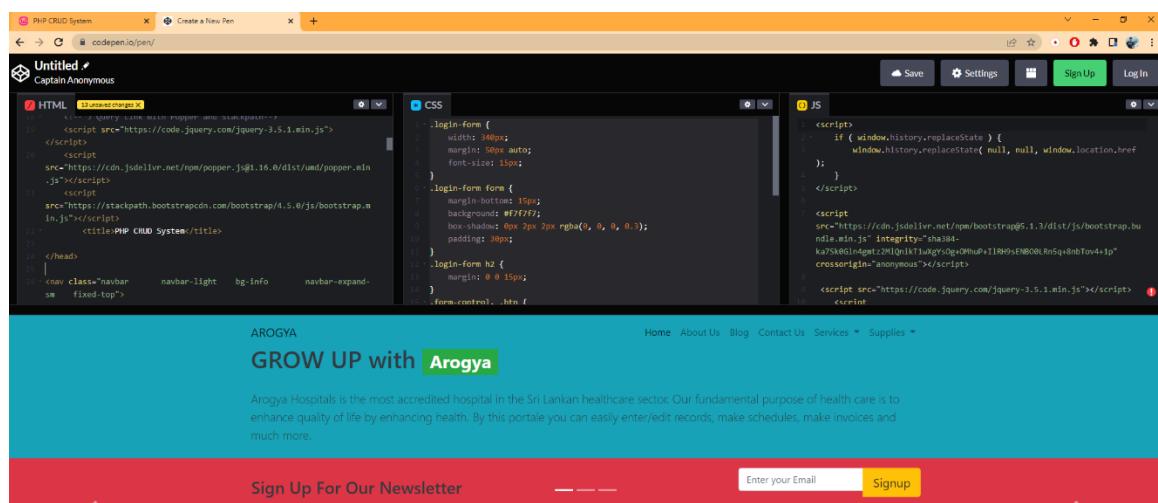


Figure 2. 13 Using Codepen for developing

5) Bootstrap

This is a front-end framework used to make web development easier and faster. It contains HTML and CSS designing templates for fonts, forms, button, tables, navigation, dialog boxes, picture carousels, and many other things. JavaScript plug-ins can also be used. It makes it easier to construct responsive designs. And since this is a free and open-source CSS framework designed for front-end web development that is responsive and mobile-first I've majorly used this for design the Hospital Management System.

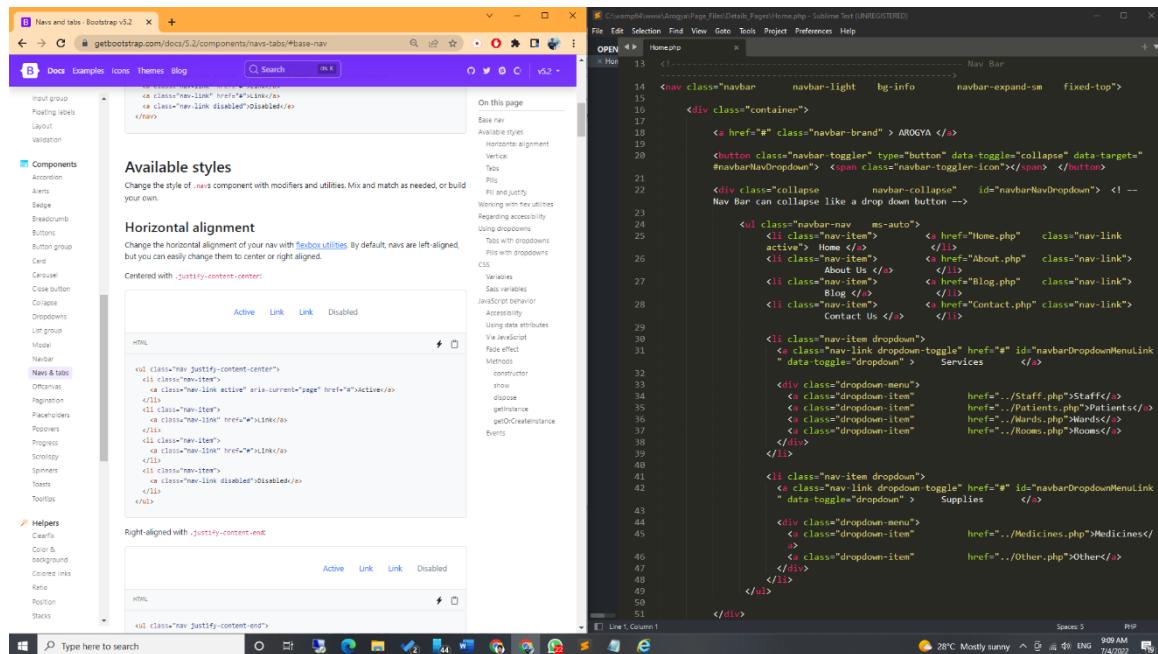


Figure 2. 14 Using bootstrap for development

Task 3

3.1 Hospital Management System Design Planning

3.1.1 Design principles to create the website

Websites that aren't well-designed perform more poorly and have Google Analytics data that aren't appropriate (high bounce rates, low time on site, low pages per visit and low conversions). There are top 10 web design principles that will make our website more visually beautiful, user-friendly, engaging, and effective.

1) Purpose

A good web design always considers the user's needs. We must assume the mindset of our website users, such as whether they are searching for information, entertainment, interaction, or to transact with our organization. Each page of our website must have a certain purpose and meet a specific requirement for our website users in the most efficient way possible.

Hence, I've clearly mentioned the purpose of this system and the clean hospital as below.

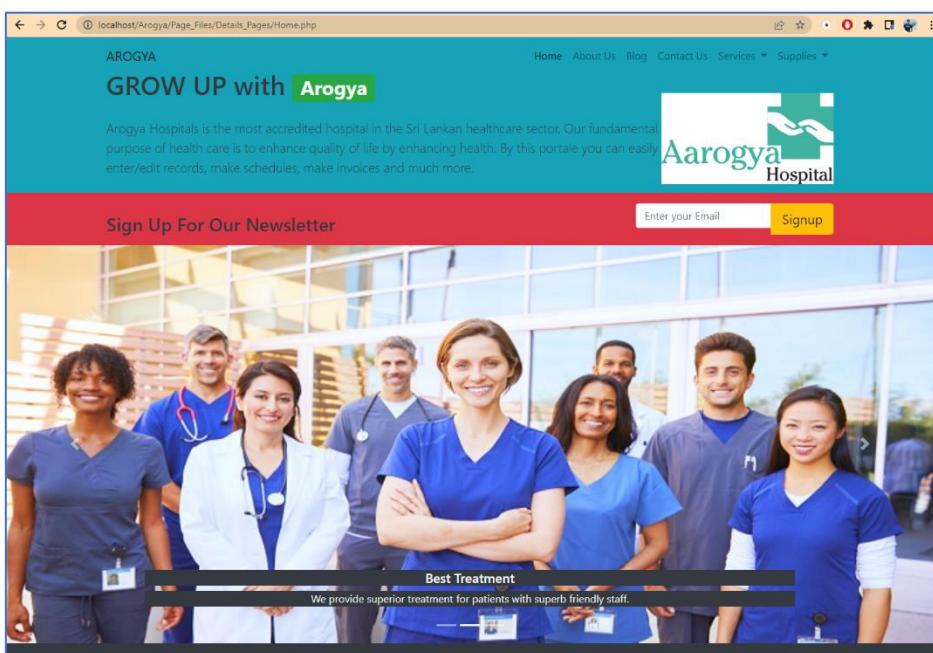


Figure 3. 1 Home Page with Purpose

And on each page, the head topic defines the purpose of the page. For example, as below, this page is specialized for managing staff, as the name implies.

| ID Number | Name | Type | Options | | |
|-----------|--------------------|------------|---------|------|--------|
| 1 | Pavel Fernando | Doctor | View | Edit | Delete |
| 2 | Ryan Wickramaratne | Nurse | View | Edit | Delete |
| 3 | Judith Michelle | Technician | View | Edit | Delete |
| 4 | Kyle Wickramaratne | Attendant | View | Edit | Delete |
| 5 | Sadew Fernando | Doctor | View | Edit | Delete |
| 6 | Mashi Perera | Nurse | View | Edit | Delete |
| 7 | Shyni Perera | Attendant | View | Edit | Delete |

Figure 3. 2 Staff Management Page

2) Communication

People on the internet prefer to get information quickly, thus it is critical to communicate effectively and make the content easy to read and digest. Some good web design strategies include using headings and subheadlines to organize content, using key points instead of long windy words, and cutting the fuzz.

Hence, I've clearly mentioned the head topics and subtopics of the page, as the below figure shows.

| Ward Number | Doctor 1 | Doctor 2 | Nurse 1 | Nurse 2 | Attendant 1 | Attendant 2 |
|-------------|----------------|----------------|--------------------|--------------|--------------------|--------------|
| Ward 1 | Pavel Fernando | Sadew Fernando | Ryan Wickramaratne | Mashi Perera | Kyle Wickramaratne | Shyni Perera |
| Ward 2 | Pavel Fernando | | Ryan Wickramaratne | | Kyle Wickramaratne | |
| Ward 3 | Sadew Fernando | | Mashi Perera | | Shyni Perera | |
| Ward 4 | Pavel Fernando | | Mashi Perera | | | |

Figure 3. 3 Ward Management Page

And in the coding, I've clearly given comments and divided each part to easily catch the purpose of the code without worrying about it. And I've organized each line according to their tags' alignment for better communication.

```

1 <!-- Footer -->
2 <?php include "../header.php"?>
3
4 <?php
5 // checking if the variable is set or not and if set adding the set data value to variable userid
6 if(isset($_GET['user_id']))
7 {
8     $userid = $_GET['user_id'];
9 }
10 // SQL query to select all the data from the table where id = $userid
11 $query="SELECT * FROM wards WHERE id = '$userid'";
12 $ward_manage= mysqli_query($conn,$query);
13
14 while($row = mysqli_fetch_assoc($ward_manage))
15 {
16     //Textbox echo id name || Database column id name
17     $id = $row['id'];
18     $wnum = $row['w_number'];
19     $doct1 = $row['Doctor1'];
20     $doct2 = $row['Doctor2'];
21     $nurs1 = $row['Nurse1'];
22     $nurs2 = $row['Nurse2'];
23     $atten1 = $row['Attendant1'];
24     $atten2 = $row['Attendant2'];
25 }
26
27 //Processing form data when form is submitted
28 if(isset($_POST['update']))
29 {
30     //Now given Variable name || Textbox id name
31     $doct1 = $_POST['doc_1'];
32     $doct2 = $_POST['doc_2'];
33     $nur1 = $_POST['nur_1'];
34     $nur2 = $_POST['nur_2'];
35     $atten1 = $_POST['atte_1'];
36     $atten2 = $_POST['atte_2'];
37
38
39 // SQL query to update the data in user table where the id = $userid
40 //Database column id name || Previously given Variable name)
41 $query = "UPDATE wards SET Doctor1 = '$doct1' , Doctor2 = '$doct2' , Nurse1 = '$nur1' , Nurse2 = '$nur2' ,
42 Attendant1 = '$atten1' , Attendant2 = '$atten2' WHERE id = '$userid'";
43 $update_user = mysqli_query($conn, $query);
44 echo "<script type='text/javascript">alert('Ward data updated successfully!')</script>";
45 header("location:../Wards.php");

```

Figure 3. 4 Ward Management Page Coding with comments

And for better communication, I've named each button with suitable icons to easily recognize the function of the button.

| ID | Name | Age | Ward | Options |
|----|--------------------|-----|--------|--------------------|
| 1 | Judith Michelle | 20 | Ward 2 | View Edit Delete |
| 2 | Ryan Wickramaratne | 25 | Ward 1 | View Edit Delete |
| 3 | Pavel Fernando | 15 | Ward 3 | View Edit Delete |
| 4 | Malshan Fernando | 25 | Ward 2 | View Edit Delete |

Patient Invoice

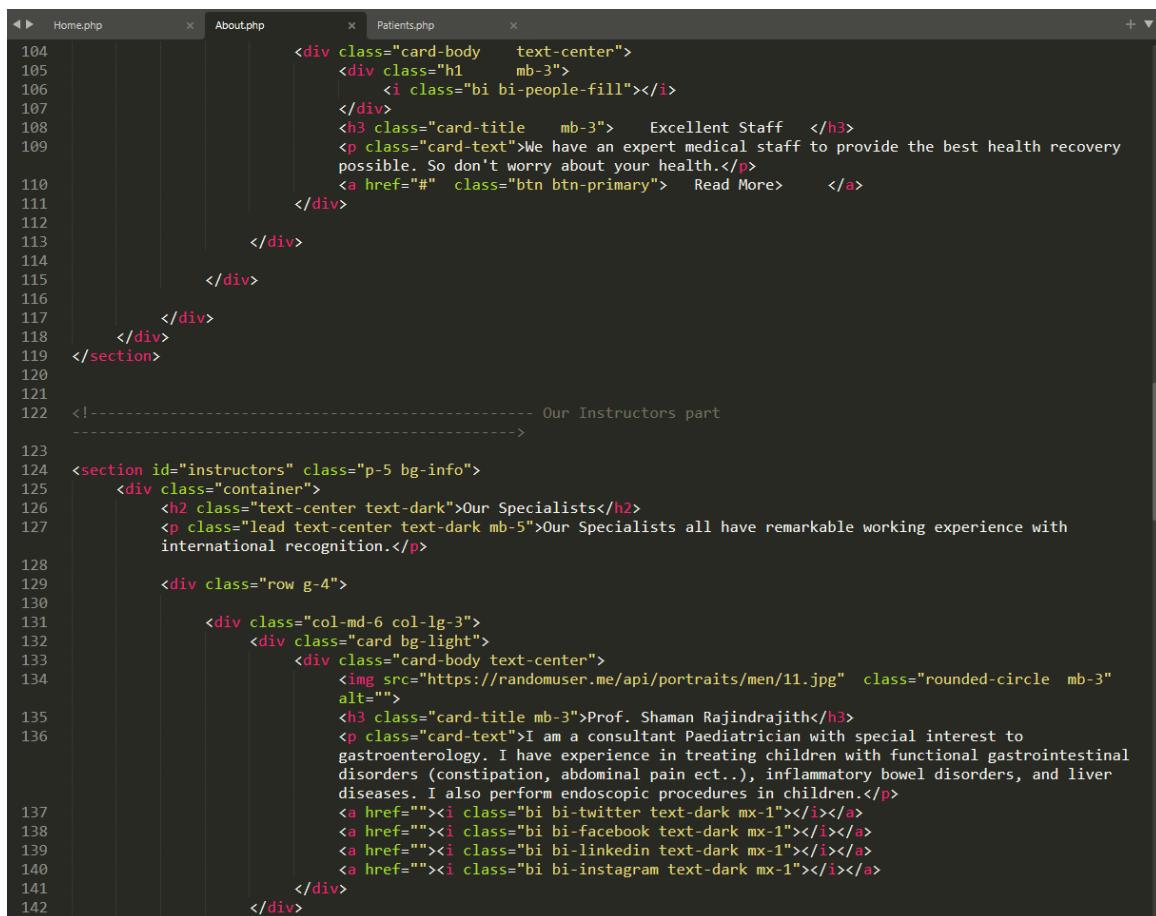
Back

Figure 3. 5 Patient Management Page buttons with icons

3) Typefaces

Sans Serif fonts, such as Lead, Arial, and Verdana, are generally simpler to read online (Sans Serif fonts are contemporary looking fonts without decorative finishes). The best font size for easy online reading is 16px, and we should limit ourselves to a maximum of three typefaces in a maximum of three point sizes to keep our design simple.

Hence I've used lead font in some parts make the font bit decorative and my this system's default font size is 16px. And I've used different heading sizes, paragraphs and much more to control the typography of my system, as the below example coding shows.



```

104 <div class="card-body text-center">
105   <div class="h1 mb-3">
106     <i class="bi bi-people-fill"></i>
107   </div>
108   <h3 class="card-title mb-3"> Excellent Staff </h3>
109   <p class="card-text">We have an expert medical staff to provide the best health recovery
110     possible. So don't worry about your health.</p>
111   <a href="#" class="btn btn-primary"> Read More </a>
112 </div>
113 </div>
114 </div>
115 </div>
116 </div>
117 </div>
118 </div>
119 </section>
120
121
122 <!-- Our Instructors part
-->
123 <section id="instructors" class="p-5 bg-info">
124   <div class="container">
125     <h2 class="text-center text-dark">Our Specialists</h2>
126     <p class="lead text-center text-dark mb-5">Our Specialists all have remarkable working experience with
127       international recognition.</p>
128
129   <div class="row g-4">
130
131     <div class="col-md-6 col-lg-3">
132       <div class="card bg-light">
133         <div class="card-body text-center">
134           
136           <h3 class="card-title mb-3">Prof. Shaman Rajindrajith</h3>
137           <p class="card-text">I am a consultant Paediatrician with special interest to
138             gastroenterology. I have experience in treating children with functional gastrointestinal
139             disorders (constipation, abdominal pain etc..), inflammatory bowel disorders, and liver
140             diseases. I also perform endoscopic procedures in children.</p>
141           <a href="#"><i class="bi bi-twitter text-dark mx-1"></i></a>
142           <a href="#"><i class="bi bi-facebook text-dark mx-1"></i></a>
143           <a href="#"><i class="bi bi-linkedin text-dark mx-1"></i></a>
144           <a href="#"><i class="bi bi-instagram text-dark mx-1"></i></a>
145         </div>
146       </div>
147     </div>
148   </div>
149 </section>

```

Figure 3. 6 About Us coding part with typefaces

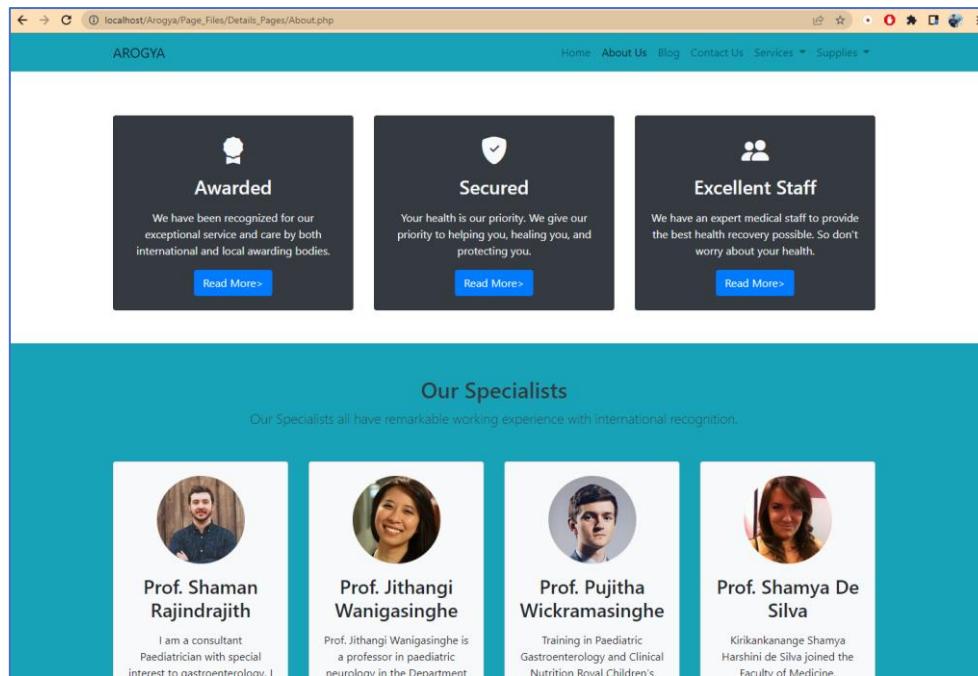


Figure 3. 7 About Us coding part's output with typefaces

4) Colours

A well-thought-out color palette can significantly improve the user experience. Colors that are complementary generate balance and harmony. Reading will be easier on the eyes if the text and background are in contrasting colors. Vibrant colors evoke feelings and should be used carefully (e.g. for buttons and call to actions). Finally, white space/negative space is particularly helpful in giving a website a fresh and uncluttered appearance.

Because of that, I've applied this design principal to make my system more efficient and attractive, as the below figure shows.

| ID | Name | Age | Ward | Options | | |
|----|--------------------|-----|--------|----------------------|----------------------|------------------------|
| 1 | Judith Michelle | 20 | Ward 2 | View | Edit | Delete |
| 2 | Ryan Wickramaratne | 25 | Ward 1 | View | Edit | Delete |
| 3 | Pavel Fernando | 15 | Ward 3 | View | Edit | Delete |
| 4 | Malshan Fernando | 25 | Ward 2 | View | Edit | Delete |

[Patient Invoice](#)

[Back](#)

Figure 3. 8 Patient Management Page with suitable colours

5) Images

A picture is worth a thousand words, and selecting the correct photos for our website can aid in brand positioning and connecting with our target audience. If we don't have high-quality professional images on hand, we might think about buying stock photos to improve the look of our website. Consider employing infographics, videos, and images as these can be far more successful at conveying than even the best written material.

Hence, I've used high-quality professional photographs and applied them to suitable topics and headings, as the following figures show.

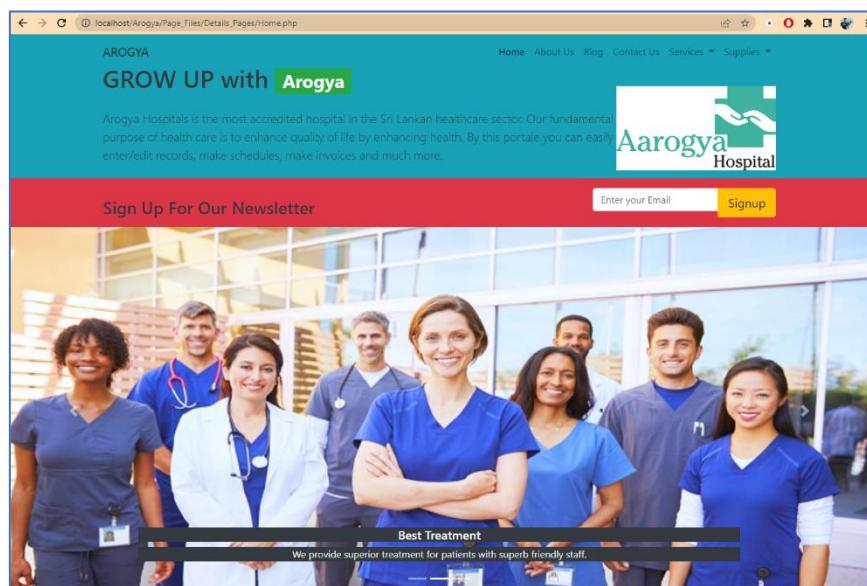


Figure 3. 9 Home page with carousel

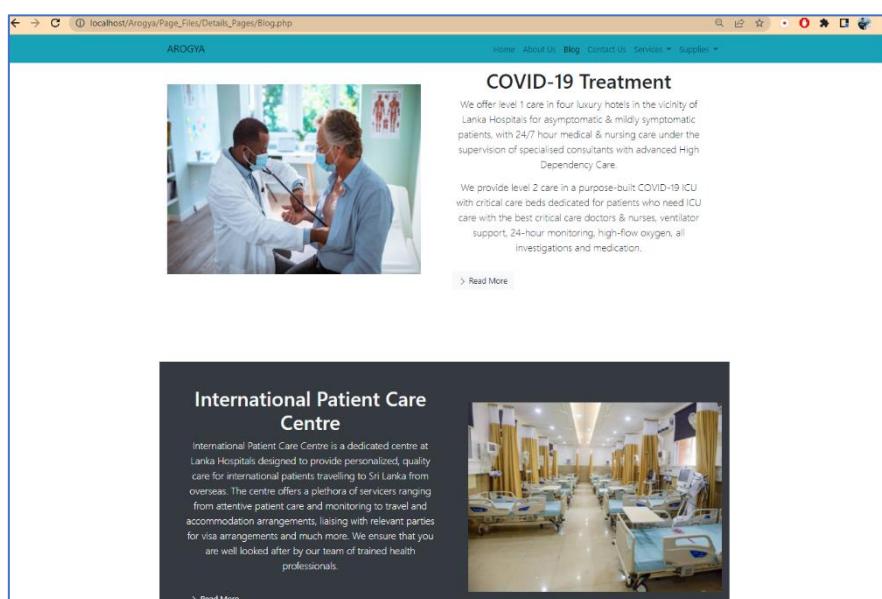


Figure 3. 10 Blog page with pictures

6) Navigation

A navigation bar assists readers in choosing categories, links, or sub-topics of interest. When using a navigation bar, users do not need to enter the URL of the specific webpage, since this is handled automatically by the navigation bar, with the navigation sections including the relevant links of the webpage.

Hence, I've used the navigation bar to simplify the browse through the system. For functional purposes, I've used drop-down menus with categorized items, as the following figures show.



Figure 3. 11 Navigation Bar of the system

7) Grid based layouts

Placing content at random on our website can result in a haphazard and chaotic appearance. Grid-based layouts divide material into sections, columns, and boxes that line up and feel balanced, resulting in a more visually appealing website design.

Hence, for proper content organization, I've used Bootstrap 12 grid system containers and tables as the below figure shows.

The screenshot shows a web browser window for 'localhost/Arogya/Page_Files/Medicines.php'. The title bar says 'localhost/Arogya/Page_Files/Medicines.php'. The page has a teal header with the word 'AROGYA' and a navigation menu with links: Home, About Us, Blog, Contact Us, Services, and Supplies (with a dropdown menu for Medicines and Other). Below the header is a section titled 'Medicine Stock Management' with two buttons: 'Add New Medicine' and 'Logout'. A table lists medicine stock details:

| ID Number | Name | Type | Quantity | Options |
|-----------|--------------|--------------|----------|--|
| 1 | Panadol | Antibiotics | 100 | View Edit Delete |
| 2 | Paracetamo | Pain Killers | 500 | View Edit Delete |
| 3 | Cough Syrup | Other | 50 | View Edit Delete |
| 4 | Gastic Syrup | Other | 50 | View Edit Delete |

Figure 3. 12 Medicines Stock Management Page

```
Medicines.php
10 <!-- Header -->
11 <?php include "../header.php"?>
12
13 <!-- Nav Bar -->
-->
14 <nav class="navbar navbar-light bg-info navbar-expand-sm fixed-top">
15
16     <div class="container" style="background-color: #f8d7da; border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;>←
17         <a href="#" class="navbar-brand" > AROGYA </a>
18
19         <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavDropdown" style="border: none; background-color: transparent; color: inherit; font-size: 1.2em; font-weight: bold; margin-right: 10px;> <span class="navbar-toggler-icon" style="font-size: 1.2em; font-weight: bold; border: none; background-color: transparent; color: inherit; width: 1.2em; height: 1.2em; border-radius: 50%; display: inline-block; vertical-align: middle; margin-right: 5px;> </span> </button>
20
21         <div class="collapse navbar-collapse" id="navbarNavDropdown" style="margin-top: 10px; margin-bottom: 10px; border: 1px solid #ccc; padding: 10px; border-radius: 10px; background-color: #fff; box-shadow: 0 0 10px 0 #ccc; position: relative; z-index: 1;> ←
22             <! -- Nav Bar can collapse like a drop down button -->
23
24                 <ul class="navbar-nav ms-auto" style="list-style-type: none; padding-left: 0; margin: 0;>
25                     <li class="nav-item" style="margin-right: 20px; border-bottom: 1px solid #ccc; padding-bottom: 5px; position: relative; z-index: 1;> <a href="Details_Pages/Home.php" class="nav-link" style="color: inherit; text-decoration: none; font-weight: bold; font-size: 0.9em; margin-bottom: 5px;> Home </a>
26                         <div style="position: absolute; bottom: -10px; left: 50%; width: 0; height: 0; border-left: 10px solid transparent; border-right: 10px solid transparent; border-bottom: 20px solid #f8d7da; margin-left: -10px; z-index: 2;> </div>
27                     <li class="nav-item" style="margin-right: 20px; border-bottom: 1px solid #ccc; padding-bottom: 5px; position: relative; z-index: 1;> <a href="Details_Pages/About.php" class="nav-link" style="color: inherit; text-decoration: none; font-weight: bold; font-size: 0.9em; margin-bottom: 5px;> About Us </a>
28                         <div style="position: absolute; bottom: -10px; left: 50%; width: 0; height: 0; border-left: 10px solid transparent; border-right: 10px solid transparent; border-bottom: 20px solid #f8d7da; margin-left: -10px; z-index: 2;> </div>
29                     <li class="nav-item" style="border-bottom: 1px solid #ccc; padding-bottom: 5px; position: relative; z-index: 1;> <a href="Details_Pages/Blog.php" class="nav-link" style="color: inherit; text-decoration: none; font-weight: bold; font-size: 0.9em; margin-bottom: 5px;> Blog </a>
30                         <div style="position: absolute; bottom: -10px; left: 50%; width: 0; height: 0; border-left: 10px solid transparent; border-right: 10px solid transparent; border-bottom: 20px solid #f8d7da; margin-left: -10px; z-index: 2;> </div>
31                 </ul>
32             </div>
33         </div>
34     </div>
35 
```

Figure 3. 13 Medicines Stock Management Page's navigation bar code part

```
Medicines.php x
70 |     ?>
71 |
72 |     <!-- Table Part
73 |     ----->
74 |
75 |     <table class="table  table-striped  table-bordered  table-hover">
76 |         <thead class="table-dark">
77 |             <tr>
78 |                 <th scope="col" class="text-center">ID Number</th>
79 |                 <th scope="col" class="text-center">Name</th>
80 |                 <th scope="col" class="text-center">Type</th>
81 |                 <th scope="col" class="text-center">Quantity</th>
82 |                 <th scope="col" colspan="3" class="text-center">Options</th>
83 |             </tr>
84 |         </thead>
85 |
86 |         <tbody>
87 |             <tr>
```

Figure 3. 14 Medicines Stock Management Page's table code part

8) “F” Pattern design

According to eye tracking studies, people scan computer screens in a "F" pattern. The majority of what people see is on the top and left sides of the screen, whereas the right side is rarely seen. Rather than attempting to force the viewer's visual flow, well-designed websites will work with the reader's natural behavior and provide content in descending order of relevance (left to right, and top to bottom).

Hence, I've used F pattern design for both coding and interfaces when designing the system, as illustrated in the above example figures.

9) Load time

Everyone despises a website that takes forever to load. Optimizing picture sizes (size and scale), consolidating code into a single CSS or JavaScript file (which lowers HTTP requests), and minifying HTML, CSS, and JavaScript are all good ways to improve page load times (compressed to speed up their load time) as I did when creating the system..

10) Mobile friendly

Since it is now usual to visit websites from numerous devices with varying screen sizes, it is critical to assess whether the website is mobile friendly. If our website is not mobile compatible, we can either rebuild it in a responsive layout (which means it will adjust to different screen sizes) or construct a specialized mobile site (a separate website optimized specifically for mobile users). That's why I used bootstrap to make this system. Because Bootstrap is a framework directed at responsive, mobile-first front-end web development,

The below figures illustrate my web page responsiveness.

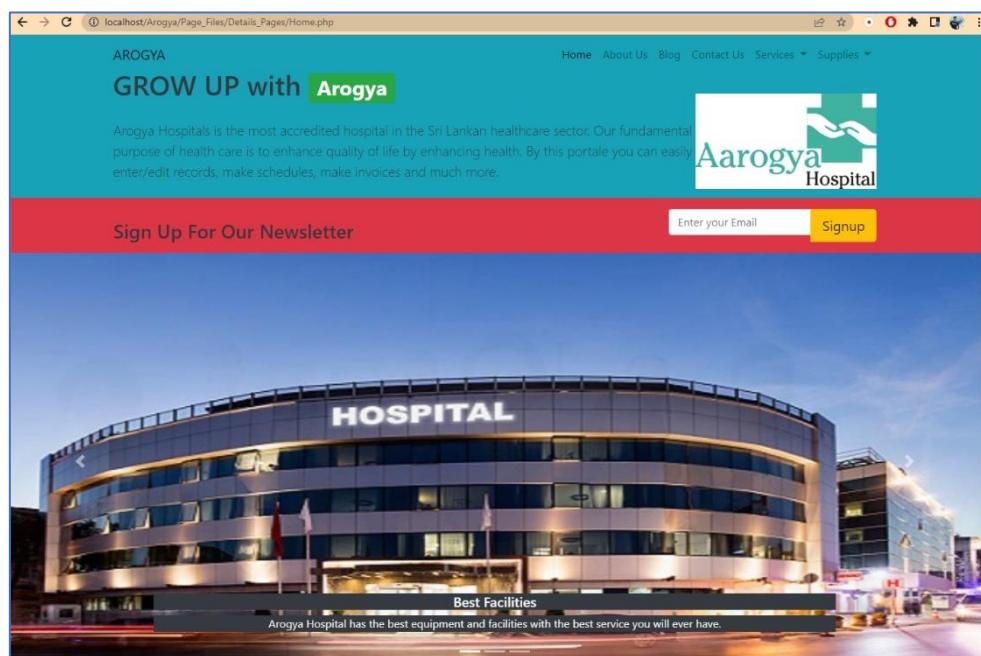


Figure 3. 15 Home page desktop mode

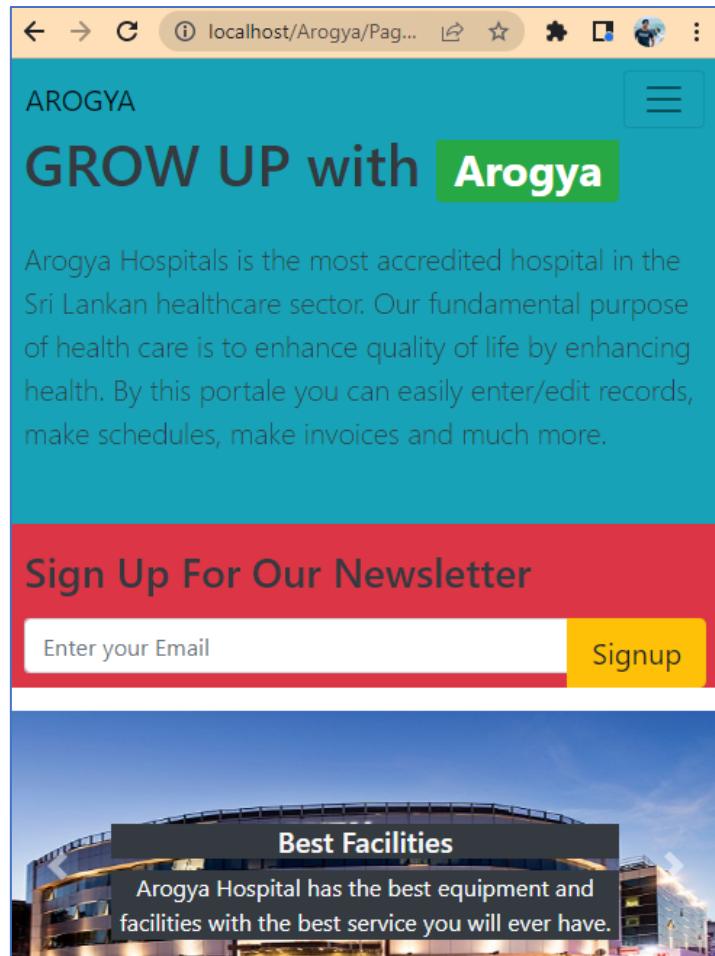


Figure 3. 16 Home page mobile mode

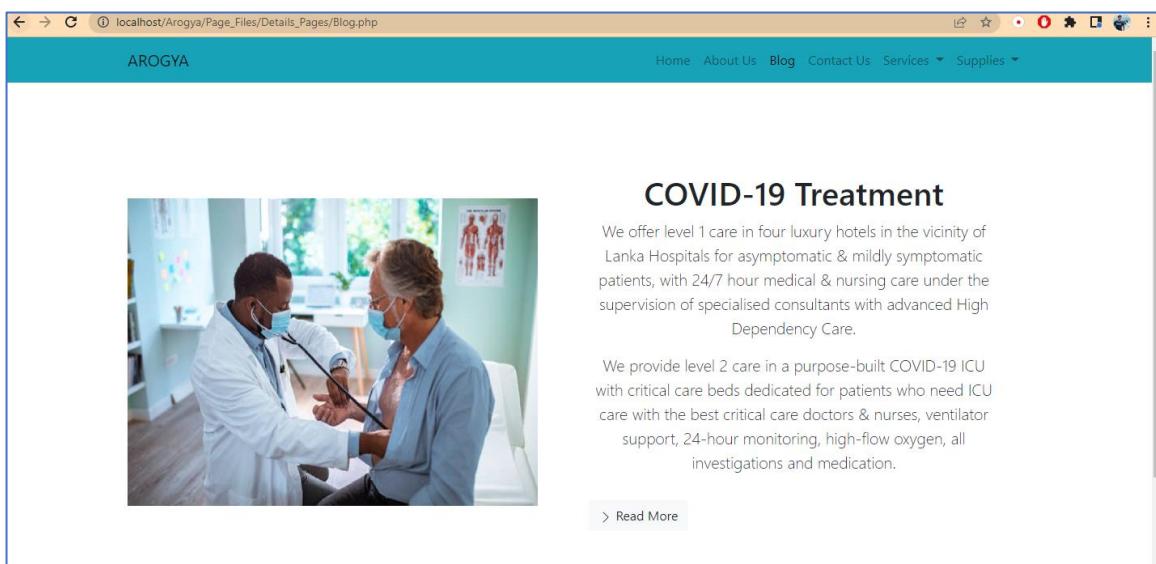


Figure 3. 17 About Us page desktop mode



Figure 3. 18 About Us page mobile mode

3.1.2 ERD with Logical Database design for the Arogya Hospital Management System

Entity Relationship Diagram (ERD) is a diagram that shows the relationship between entity sets contained in a database. In other words, ER diagrams help in the explanation of database logical structure. Entities, attributes, and relationships are the three main components that ER diagrams are built on.

Rectangles are used to represent entities, ovals are used to define attributes, and diamond shapes are used to show relationships in ER Diagrams.

An ER diagram appears to be quite similar to a flowchart at first glance. The ER Diagram, on the other hand, contains many specific symbols, and the meanings of these symbols distinguish this model. The entity framework infrastructure is represented by the ER Diagram.

The uses of ER Diagrams as below.

- ERD supports in the define the concepts used in entity relationship modelling.
- These diagrams show how all of our tables should relate, as well as what fields will be on each table.
- This helps in the definition of entities, attributes, and relationships.
- ER diagrams may be converted into relational tables, allowing us to create databases quickly.
- Database designers can utilize ER diagrams as a roadmap for implementing data in specific software applications.
- With the help of an ER diagram, the database designer gains a better knowledge of the information that will be stored in the database.
- The ERD Diagram helps us to explain the database's logical structure to users.

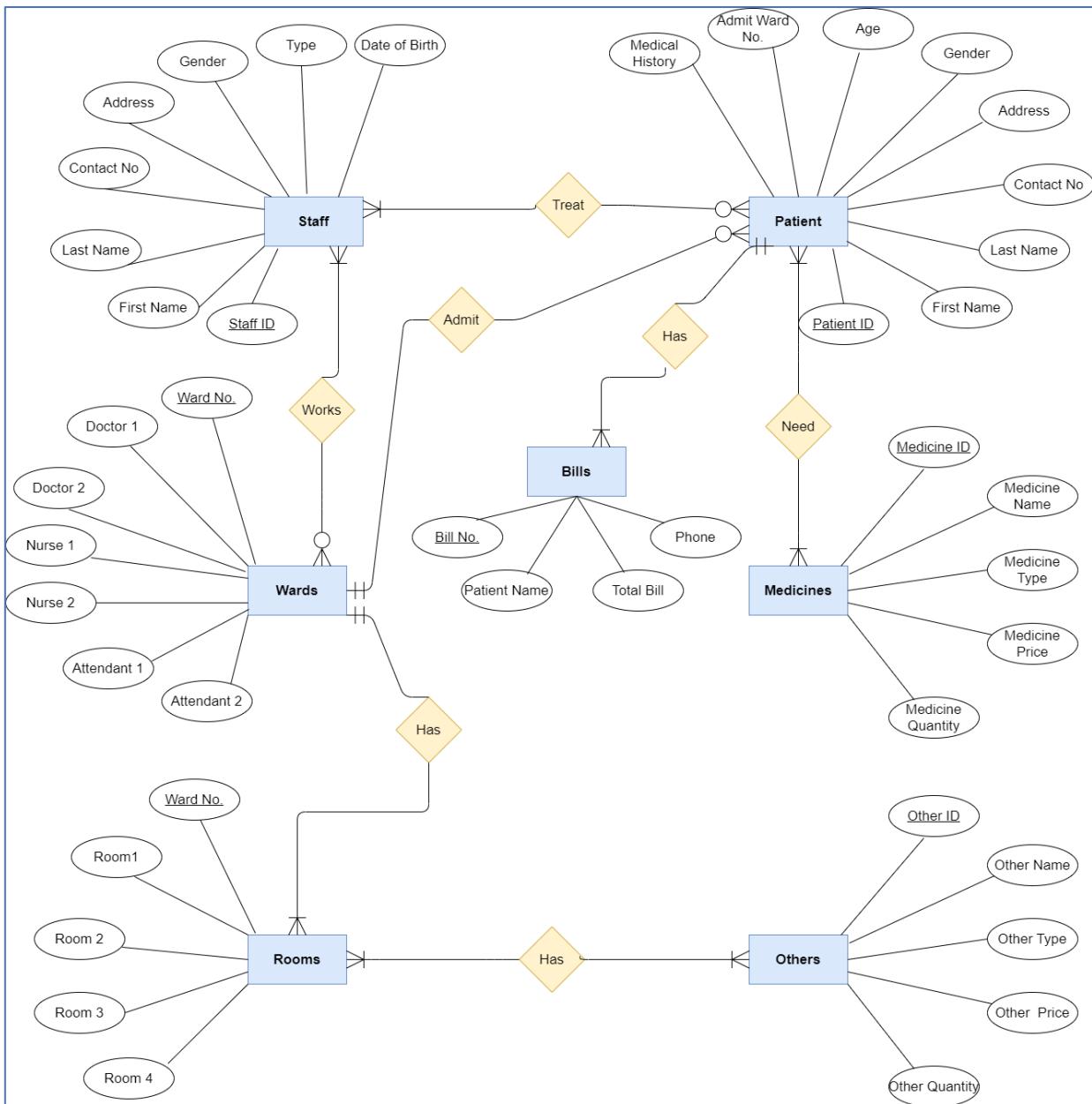


Figure 3. 19 ER Diagram for Arogya Hospital Management System.

The entities and relationships represented in an ERD are focused on the requirements of the business. The necessity to satisfy database design has not yet been considered. The real design of a database is represented by a logical database. It is concerned with the transformation of a logical design into a schema-level design that will be turned into a relational database.

When modelling a logical database, ERD is used as the foundation, and then primary keys, foreign keys, and constraints are defined. Relationships are sometimes resolved by adding new tables, such as a Linked table for a many-to-many relationship.

Because the ERD and logical database represent the business requirement and database schema, comparing them both helps to identify the differences, ensuring that the database follows the initial business's user requirements regardless of modifications.

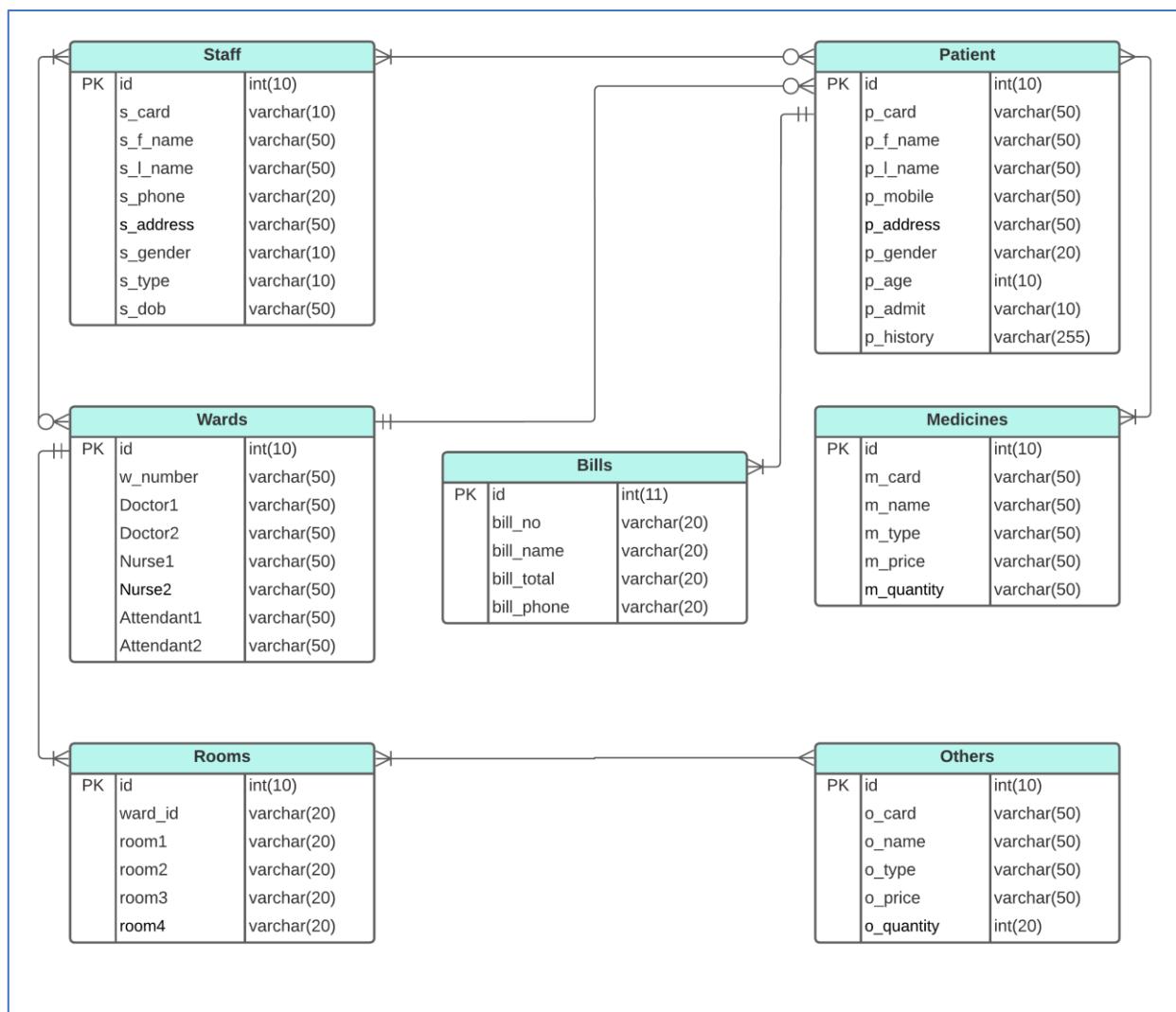


Figure 3. 20 Logical Database Design for Arogya Hospital Management System

3.1.3 Class Diagram for the Arogya Hospital Management System

The Unified Modeling Language (UML) can assist us in modeling systems in a variety of ways. The class diagram is one of the more common UML types. Class diagrams are a form of structural diagram that are used by software engineers to document software architecture. They specify what must be included in the system being modeled.

UML was created as a standard model for describing an object-oriented programming method. Class diagrams are the building blocks of UML because classes are the building blocks of objects. A class diagram's many components can represent the classes that will be programmed, the principal objects, or the interactions between classes and objects.

The shape of the class is a rectangle with three rows. The top row carries the class's name, the middle row contains the class's characteristics, and the bottom part expresses the methods or operations that the class may employ. Classes and subclasses are placed together to demonstrate the object's static relationship.

Class diagrams provide numerous advantages to any organization. UML class diagrams can be used to:

- Draw data models for information systems, no matter how simple or complex they are.
- Improve the understanding of an application's overall schematics.
- Visually articulate any system-specific requirements and convey that knowledge throughout the organization.
- Create thorough charts that indicate any specific code that must be designed and implemented in accordance with the defined structure.
- Give an implementation-independent description of types that are employed in a system and are later passed between its components.

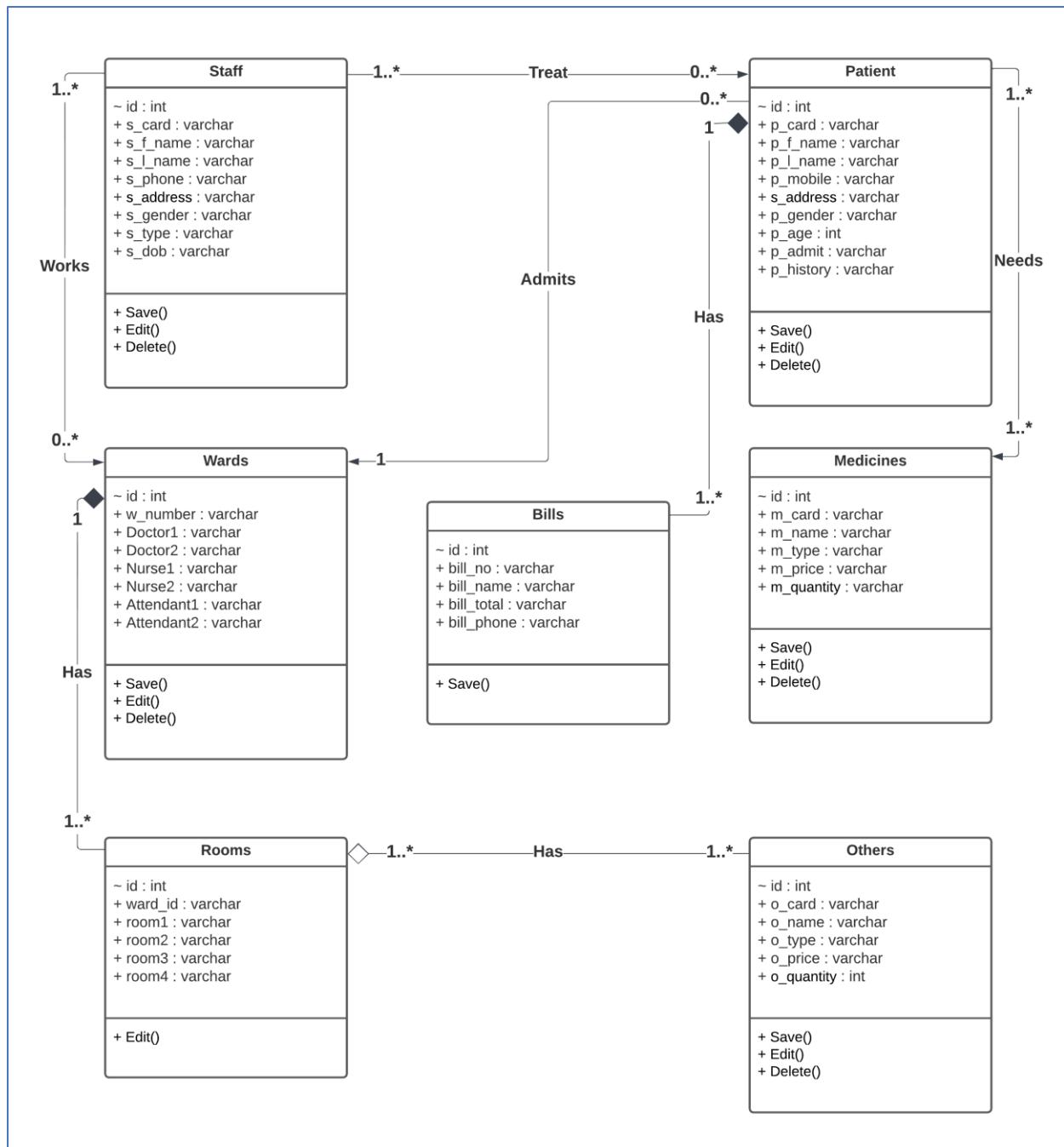


Figure 3. 21 Class Diagram for Arogya Hospital Management System

3.1.4 Use Case Diagram for the Arogya Hospital Management System

A use case diagram in the Unified Modeling Language (UML) can summarize the details of our system's users (also known as actors) and their interactions with the system. We employ a set of specific symbols and connectors to create one.

A good use case diagram can assist our team in discussing and representing following things:

- Interactions between our system or application and individuals, organizations, or external systems.
- Goals that our system or application assists those entities (referred to as actors) in achieving.
- Our system's capabilities.

The following scenarios benefit from UML use case diagrams:

- Defining the objectives of system-user interactions
- Defining and structuring system functional requirements
- Defining a system's context and requirements

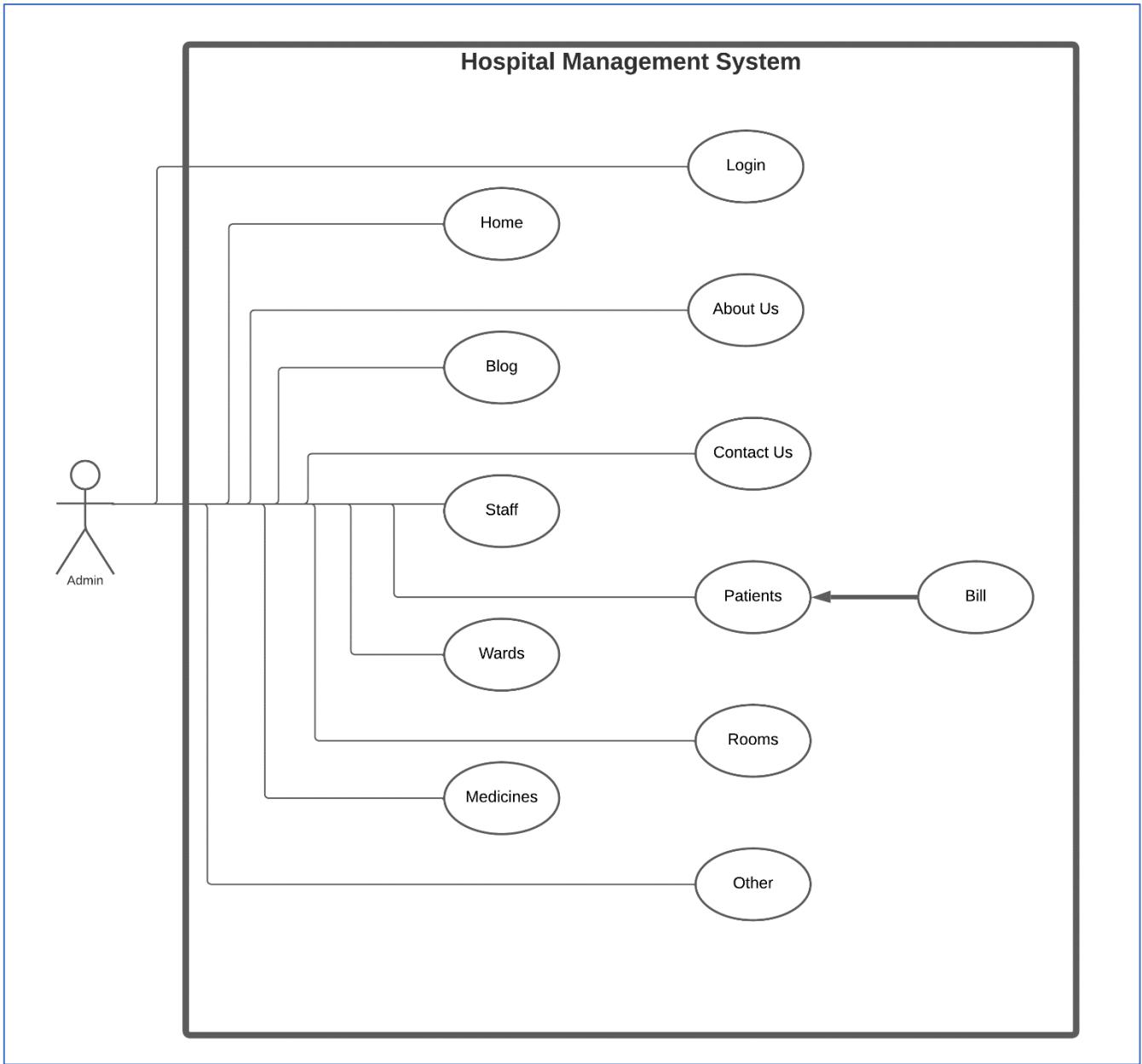


Figure 3. 22 Use Case Diagram for Arogya Hospital Management System.

3.2 Types of Requirements in System development

3.2.1 Introduction

Every software project needs a complete set of requirements. High-level requirements specify the demands and goals of the product's business. They also clearly specify what stakeholders expect given of features, functionality, behaviors, and performance.

Software requirements are a method of identifying and clarifying the why, what, and how of a business's application. Software requirements can serve as a roadmap to help a developer construct the right product fast and with the least amount of expensive rework when they are well documented. The audience and project maturity determine the precise types of software requirements and documents an IT organization creates for a given project. Organizations actually frequently create multiple requirements documents, each customized to the particular requirements of business executives, project managers, and application developers.

3.2.2 Business Requirements

A Business Requirements is the initial phase of a product's life cycle and is also referred to as a business needs specification. By methodically listing high-level business requirements in relation to customers' wants, it describes the issues that a product, service, or system is attempting to address. It lists things the project should offer in addition to non-negotiables, which can be seen as objectives for the development team.

Example of Business Requirement used for this System :-

- Method to submit records to database.
- Method to login to system for authorized people.

3.2.3 Functional Requirements

Functional Requirements logically describes how a project or system will fulfill the requirements stated in the business requirements document. By capturing the expected behavior of the system, which is described as services, tasks, or functions that the developers have committed to offer, it specifies the functionality of the system in detail. An FRD focuses on what people might see while engaging with the system rather than defining the "inner-workings" and standards.

Examples of Functional Requirements used for this System:-

- The user is taken back to the main window after clicking the Submit button, and the dialog is closed while inserting the records into the database.
- When user give correct credential details, he/she should have taken to home page of the system.

3.2.4 Quality Requirements

The customer's expectations for the final product's quality are described in the quality requirements document. It comprises of a number of requirements, considerations, and measurements that must be addressed. Reliability, consistency, availability, usability, maintainability, and customer experience may be emphasized in quality requirements.

Examples of Functional Requirements used for this System:-

- The system should be available anytime of the day.
- Until the admin makes a change, the data that was placed into the database should remain constant.

3.2.5 User Requirements

The term "user requirements" refers to a set of user standards that inform software about what the device should be able to do. The user specifications specification should be free of solution-oriented bias and use terms that are relevant to the user's problem area.

According to this given project Arogya is a hospital, situated in Sri Lanka. For the time being, they use a manual approach to handle and maintain vital data. The existing method requires dozens of new paper forms, with data sources scattered over the hospital's administration infrastructure. Data which are on paper forms are frequently missing or fails to meet management requirements. In the Arogya hospital, there are many copies of the same information, which might lead to data inconsistencies across different data storage.

As a result, this client requires keep track of patient personal information and medical history, as well as staff information, room and ward scheduling, staff scheduling, multiple facility waiting lists, and a variety of other documents. And they want to do all of these in low-cost manner.

The user requirements that needed to apply to the Arogya Hospital system are summarised below.

- 1) Need to system to handle paper records (For admins).
- 2) Need this system for low-cost manner.
- 3) Need an efficient system.
- 4) Need to record patient information with medical history.
- 5) Need to record staff information with staff scheduling.
- 6) Need to check room availability.
- 7) Need to create patient invoices.
- 8) User should have ability to move through forms easily

In addition, the system will have to adhere only to the following user group:

1. Admin

3.2.6 System Requirements

System requirements are the minimum criteria that a device must meet in order to use specific hardware or software. These are usually provided by the system developer to the consumer. As my knowledge Polly Pipe must meet the following system requirements in order to design and implement a database with a system.

First there should have Entity Relationship diagram tool to get visual starting point for database architecture. And also, ERD determining information system requirements across an organization. For that I'm choosing “draw.io” application. Draw.io is a free diagramming application. It can be used to create flowcharts, network diagram software, ER diagrams, and can use to create database schema.

Then to get better understand of the created ERD there must have Logical database designer tool. For that I'm using “Lucidchart” web application. Lucidchart allows users to create and share professional flowchart diagrams for a variety of purposes, including brainstorming and project management. And with this application it's very easy to create logical database rather than using “draw.io” application. Since this is free and cloud-based application, can save work anytime anywhere.

To host the system, I'm using WAMP server. It's a software stack, which implies that installing WAMP on the operating system installs Apache, MySQL, and PHP (Windows in the case of WAMP). Despite the fact that they can be installed independently, they are frequently bundled together, and for good reason. And it is localized on our system and not connected to the internet, it allows us to test all web page functionalities without risk. First and foremost, this eliminates the need to wait for data to be transferred to our site, and it also makes backups easier.

For coding part I've used Sublime Text. Sublime Text is a powerful text editor that is popular among programmers. Syntax Highlighting, Auto Indentation, File Type Verification, and Packages are just a few of the features that make working with code easier.

Then to create database for the system I selected MySQL since it can be used to quickly create or modify SQL database. MySQL is the popular database of choice for PHP programmers. MySQL database is an open-source Relational Database Management System (RDBMS) that employs SQL to facilitate data retrieval and give excellent support for PHP MySQL web application development.

The system requirements (Software Requirements) that applied to the Arogya Hospital Management System are summarised below.

1. Draw.io application to create ERD.
2. LucidChart application to create logical database.
3. WAMP server to host website system.
4. Sublime text for text editor
5. MySQL to handle database.
6. Windows operating system (Windows 8/10/11)

Hardware Requirements :-

- Any basic computers which having Windows Operating System.
- Internet connection should have at least normal speed (10mbps or more).

3.2.7 Evaluate the effectiveness of designed forms with identified user requirements

Table 3. 1 User requirement and Effectiveness of designed forms to fulfill the user requirement

| User Requirements | |
|---|---|
| User Requirement key point | Effectiveness of designed forms to fulfill the user requirement |
| Need to system to handle paper records. | <p>For this, I've added a logging system for admins. The user wants to handle their paper records as per the user's requirements. So, the system is only needed for hospital management. Not for patients or customers, so I created a logging page with a predefined username and password that will only be shared among system users.</p> <p>Admin can log into system and can the paper records to the system and it'll never lost again.</p> |
| Need this system for low-cost manner. | <p>The client is willing to pay a low cost for this system. Hence, adding unrelated content creates redundancies. So I created this system for only admins (Hospital Management). Making a system for both admins and customers could be highly costly since a lot of work has to be done to make a system for both parties.</p> |
| Need an efficient system. | <p>Despite the user's need for a low-cost system, they are asking for a system with efficiency. So to make the system much more efficient, I've followed some SEO tactics and coding strategies to make this system efficient.</p> |

| | |
|--|---|
| | <p>Rather than creating the system on a single page, I've divided the contents and added a navigation bar to surf through them easily.</p> <p>Then I've added separate View, Edit, and Delete functions and followed every design principle as mentioned before to make the system attractive, simple to use and more efficient for the user.</p> <p>I've added more pictures, icons, headings, comments and many more elements to improve communication between users and the system.</p> <p>I've made some interconnections between some pages to handle the records easily in the system. For example, in the Ward Management Page when handling Ward Staff, the inserted staff records through the staff page can be retrieved from this page. In the combo box, we can see the staff records in the Ward Management page. Hence, the user does not need to go through the Staff Records page again when scheduling Ward staff into the system, which makes this system more efficient.</p> <p>And in each record interface tables contain necessary details in every row. For example, in Staff Management interface page table showing only Staff ID, Name and Type. But to get more details of them we can use View button next to every record to see much more details which influence to make this system more efficient.</p> |
|--|---|

| | |
|--|---|
| Need to record patient information with medical history. | <p>In the Patient Management page, the admin can insert new patient records, edit existing records or delete them as required.</p> <p>These records are saved on the MySQL server. Hence, data can be secured in the cloud rather than storing them in physical storage devices.</p> |
| Need to record staff information with staff scheduling. | <p>On the Staff Management page, the admin can insert new staff records, edit existing records or delete them as required.</p> <p>This inserted data can be retrieved on the Ward Management page when scheduling staff to wards. This improves the effectiveness of this system.</p> <p>These records are saved on the MySQL server. Hence, data can be secured in the cloud rather than storing them in physical storage devices.</p> |
| Need to check room availability. | <p>On the Room Management page, the admin can check the room availability. And the admin can change the room availability as well.</p> <p>And the room count can be retrieved from the Ward Management table. So on this page, the user doesn't need to go back to the Ward Management page and check for room count since the room count is automatically retrieved from the Ward Management table.</p> |

| | |
|---|---|
| Need to create patient invoices. | <p>Making invoices is very important for every organization. Since this is a hospital management system, they only charge money from patients. Hence, rather than creating another page for invoices, I've created an invoice modal in the Patient Management page.</p> <p>Hence, the admin can easily add invoices for the patients on the patients' page and this data will be recorded in the MY SQL. But the invoice records that have been saved in the system can't be edited or deleted by the admins of the hospital to make sure the billing records are safe. So these records can be viewed, edited, or deleted by the server-side admin or by the host.</p> |
| User should have ability to move through forms easily | <p>To make it easier to navigate through each page, I've introduced a navigation bar for each page. And I've fixed the navigation bar at the top, so it will always appear to the user, whatever page they are on.</p> <p>And I've added Staff, Patients, Wards, and Rooms pages links in a Services combo box in the navigation bar to make it simpler. I also added Medicine and Other page links in the Supplies combo box to help users navigate through the system easily and to avoid confusion.</p> |

3.2.8 Evaluate the effectiveness of designed forms with identified system requirements

Table 3. 2 System requirement and Effectiveness of solution to fulfill the system requirement

| System Requirements | | |
|--------------------------|--|---|
| System Requirement | Solution | Effectiveness of solution to fulfill the system requirement |
| Create ERD. | Draw.io application to create ERD. | <p>Since Draw.io is a free diagramming application it can be used to create flowcharts, network diagram software, ER diagrams, and can use to create database schema.</p> <p>By using this platform, I was able to create ERD with multiple shapes easily. And connecting and disconnecting each attribute, entity and relation with lines and arrows was easy.</p> <p>If I want to move part of the design, only I had to do was selecting the portion of the design which I want to move. This helped me to save time rather than recreating the design from the start.</p> |
| Create logical database. | LucidChart application to create logical database. | For this I've used "Lucidchart" webapplication. Lucidchart allows users to create and share professional flowchart diagrams for a variety of purposes, including brainstorming and project management. |

| | | |
|--------------------------------|-------------------------------------|--|
| | | <p>And with this application it was very easy to create logical database rather than using “draw.io” application. Since this is free and cloud-based application, I was able to save work anytime anywhere.</p> <p>If I want to move part of the design, only I had to do was selecting the portion of the design which I want to move like I did in Draw.io application. This helped me to save time rather than recreating the design from the start.</p> <p>And this web application helped me to save time connecting and reconnecting each logical tables easily to show the relations. And displaying Primary Keys and Foreign Keys and interconnecting them was so efficient with this app.</p> |
| Host the multipage web system. | WAMP server to host website system. | <p>To host the system, I've used WAMP server. It's a software stack, which implies that installing WAMP on the operating system installs Apache, MySQL, and PHP (Windows in the case of WAMP).</p> <p>Wamp server is a collection of programs that may be used to convert a standard desktop PC into a fully functional web server that supports HTTP, PHP, MySQL, PHPMyAdmin, and SQLBuddy. The benefit</p> |

| | | |
|------------------|------------------------------|--|
| | | <p>is that it is simple to configure using the built-in tools.</p> <p>It's also set up in such a way that everything I need is just a click away. The configuration panels provide additional details about how settings should be configured.</p> <p>Another advantage is that the packaged programs are all automatically set to work together. All I have to do is unpack script, database, or application, run the installer (if any).</p> |
| For code editing | Sublime text for text editor | <p>Sublime Text does an excellent job with color coding and syntactic definitions, making code very understandable. The tab selection is excellent; it's simple to identify which files have been modified and to compare numerous files.</p> <p>Auto-complete is remarkably accurate and effective.</p> <p>Emmet is a must-have add to Sublime Text's toolkit for front end developers and anyone else who spends a lot of time dealing with HTML and CSS. This package allows us to type dynamic CSS expressions, sometimes known as 'abbreviations,' and have them expand into fully functional code.</p> |

| | | |
|-------------------------------------|---------------------------|--|
| Handle information in the database. | MySQL to handle database. | <p>Then to create database for the system I selected MySQL server since it can be used to quickly create or modify SQL database. And also, it can be used to add database objects including tables, views, and stored procedures. And it was very easy to connect web application.</p> <p>MySQL is the most secure and dependable database management system available, and it's utilized in popular web applications like WordPress, Drupal, Joomla, Facebook, and Twitter. The data security and transactional processing capability that come with the latest version of MySQL can be quite beneficial to any organization, particularly if it is an eCommerce business with frequent money transfers.</p> <p>MySQL has its own storage-engine structure, which makes it easier for system administrators to set up the MySQL database server for optimal performance. MySQL is built to fulfill even the most demanding applications while assuring optimum speed, full-text indexes, and distinct memory caches for increased performance, whether it's an eCommerce website that receives a million queries per day or a high-speed transactional processing system.</p> |
|-------------------------------------|---------------------------|--|

3.3 Hospital Management System Outputs

3.3.1 Initial Design (Wireframes)

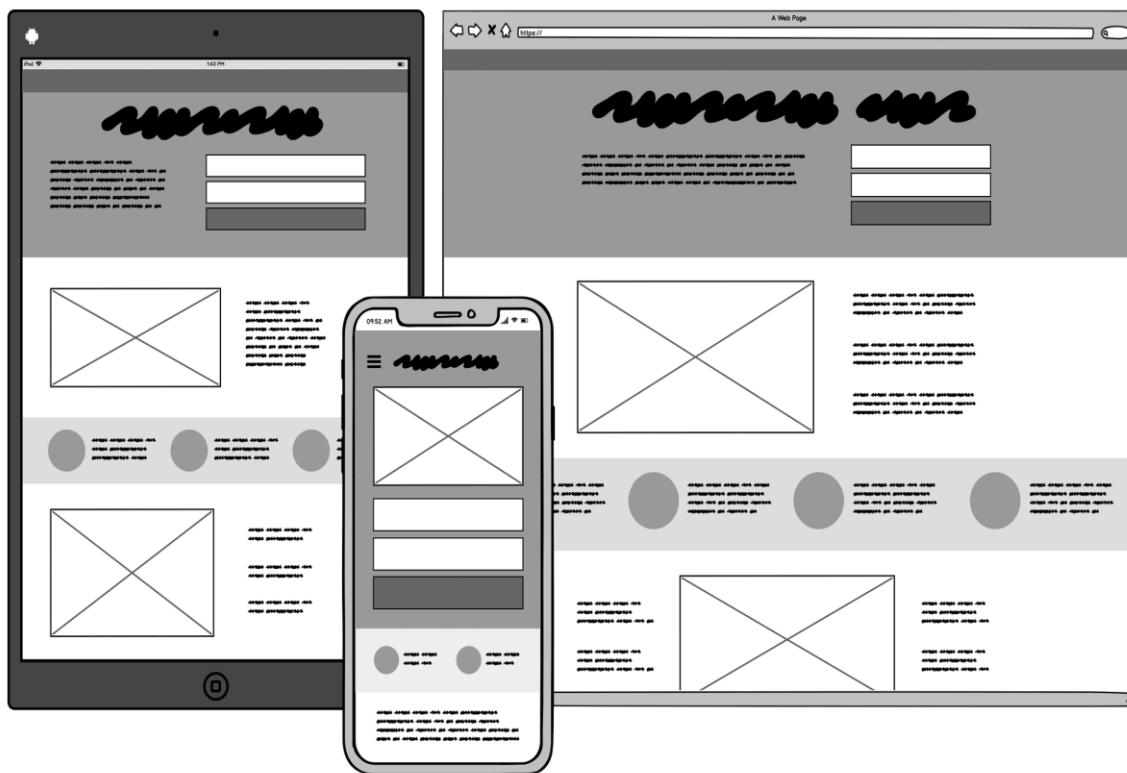


Figure 3. 23 Wireframe samples

A wireframe is a sketch or blueprint that is beneficial for facilitating communication and thought among programmers and designers regarding the organizational layout of the software or website they are developing.

There are numerous methods to construct the same screen, but only a select handful will effectively communicate our message and produce a user-friendly piece of software or website. The most crucial aspect of software design may be establishing a solid interface structure. We will save a ton of time and hard rework if we complete this job immediately, before any code is written and before the visual design is complete.

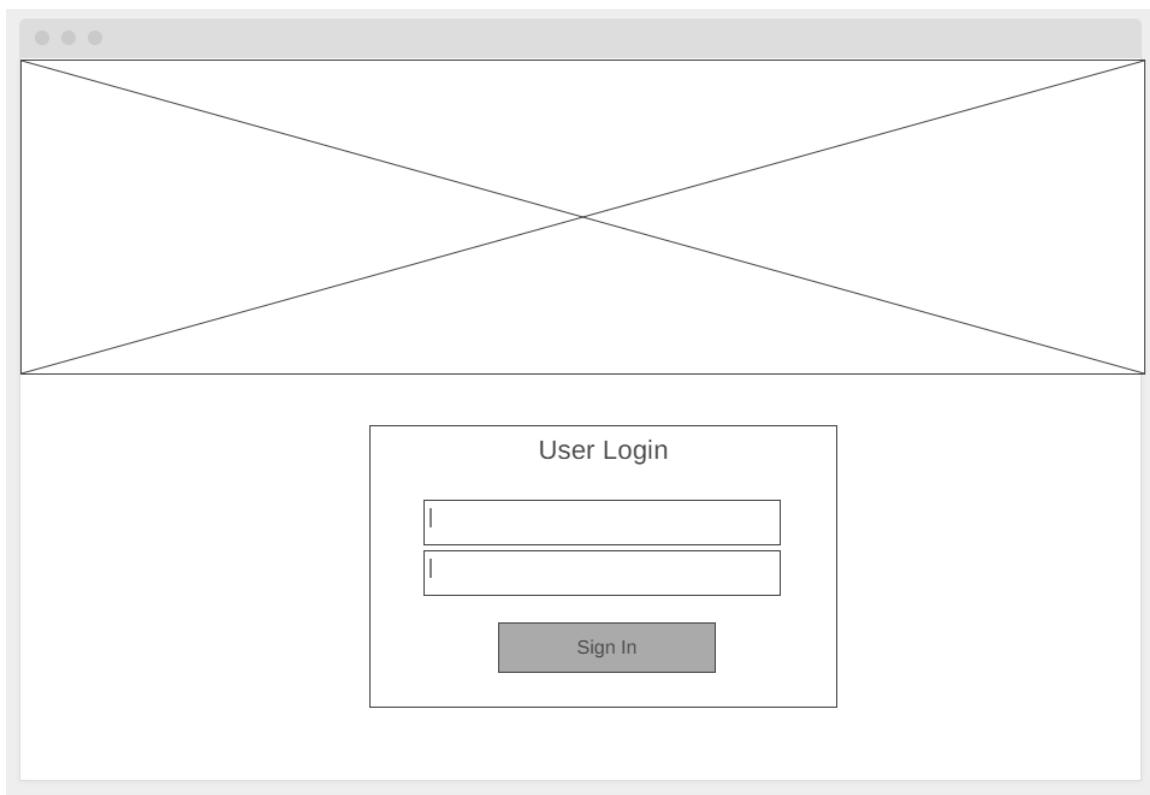


Figure 3. 24 Login Page Wireframe

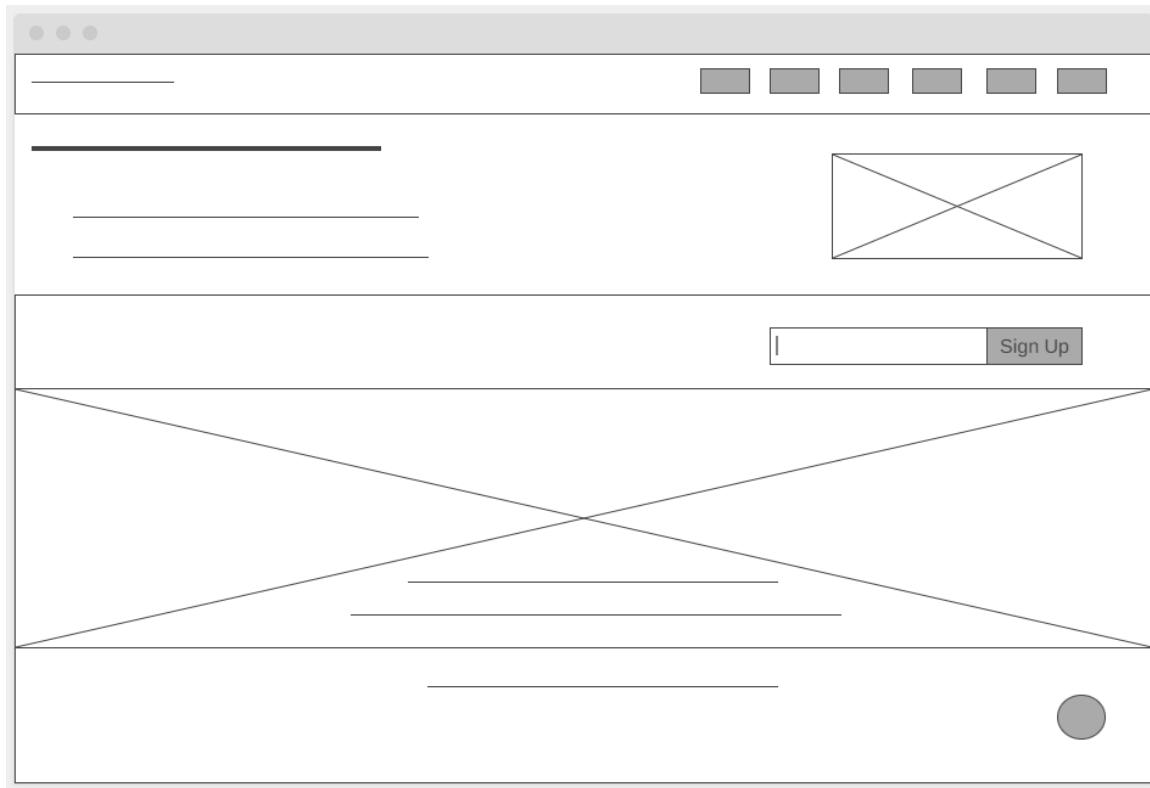


Figure 3. 25 Home Page Wireframe

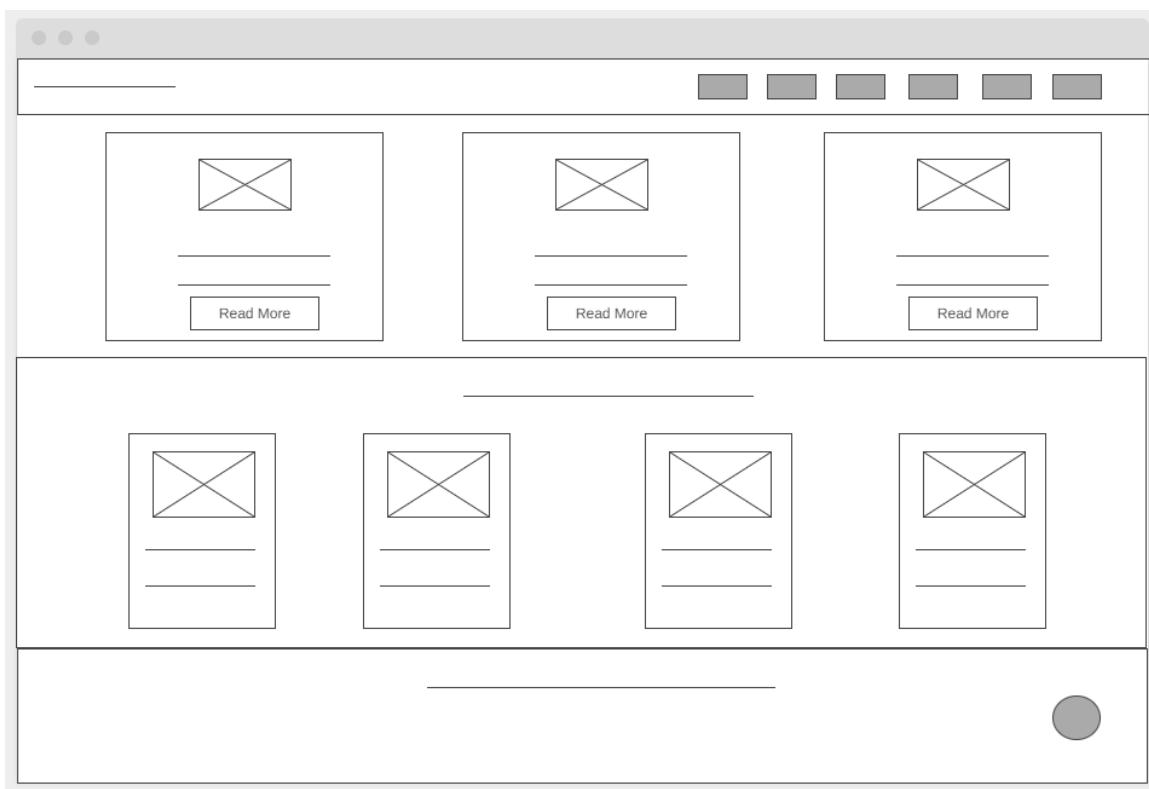


Figure 3. 26 About Us Page Wireframe

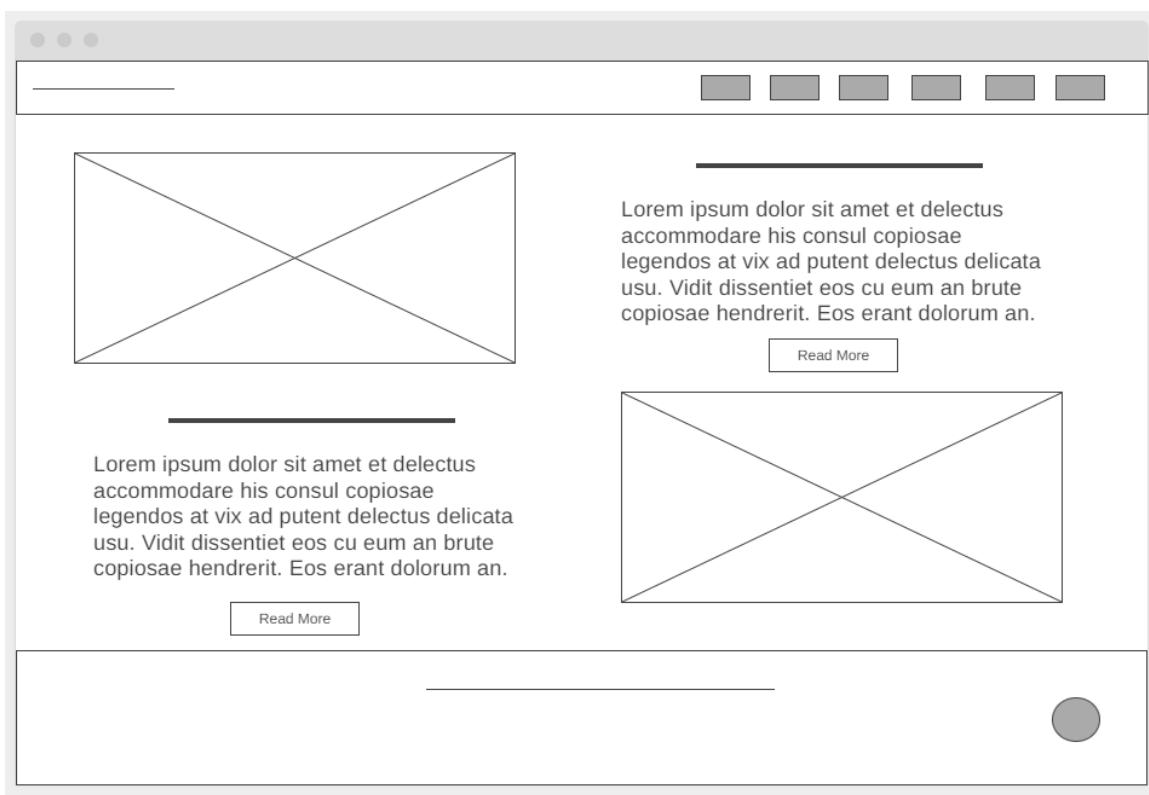


Figure 3. 27 Blog Page Wireframe

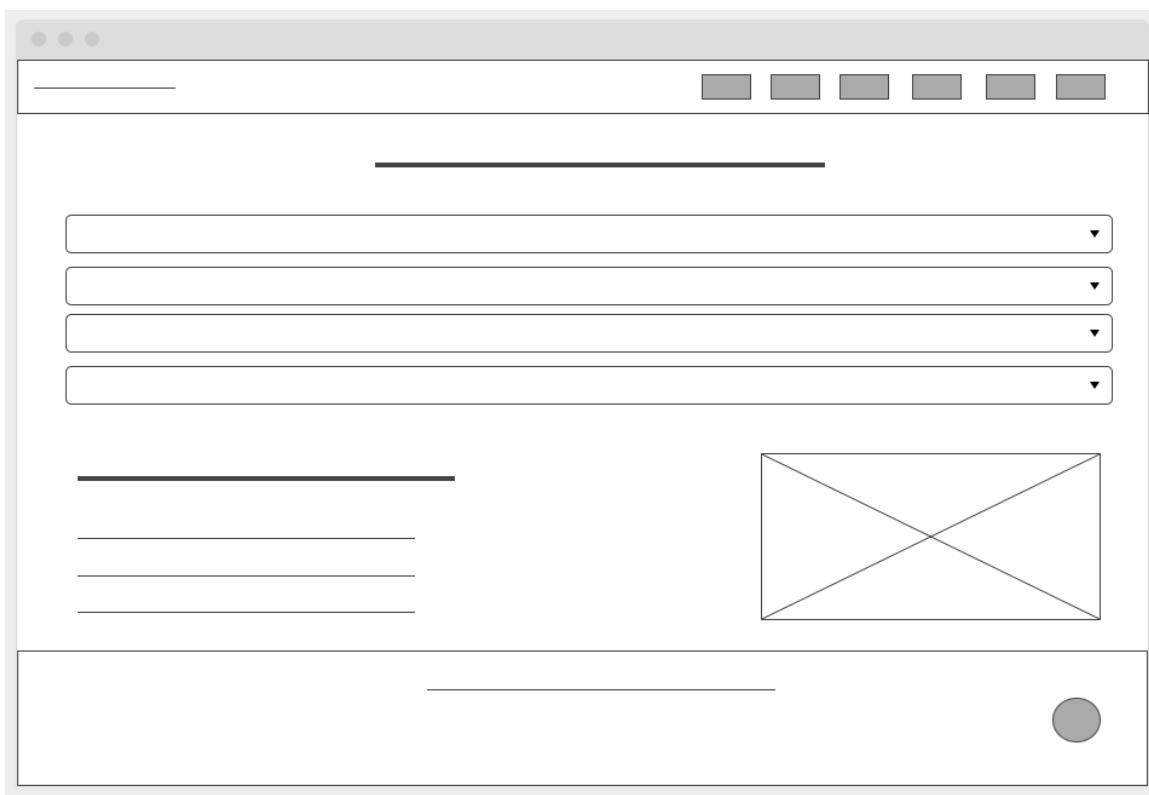


Figure 3. 28 Contact Us Page Wireframe

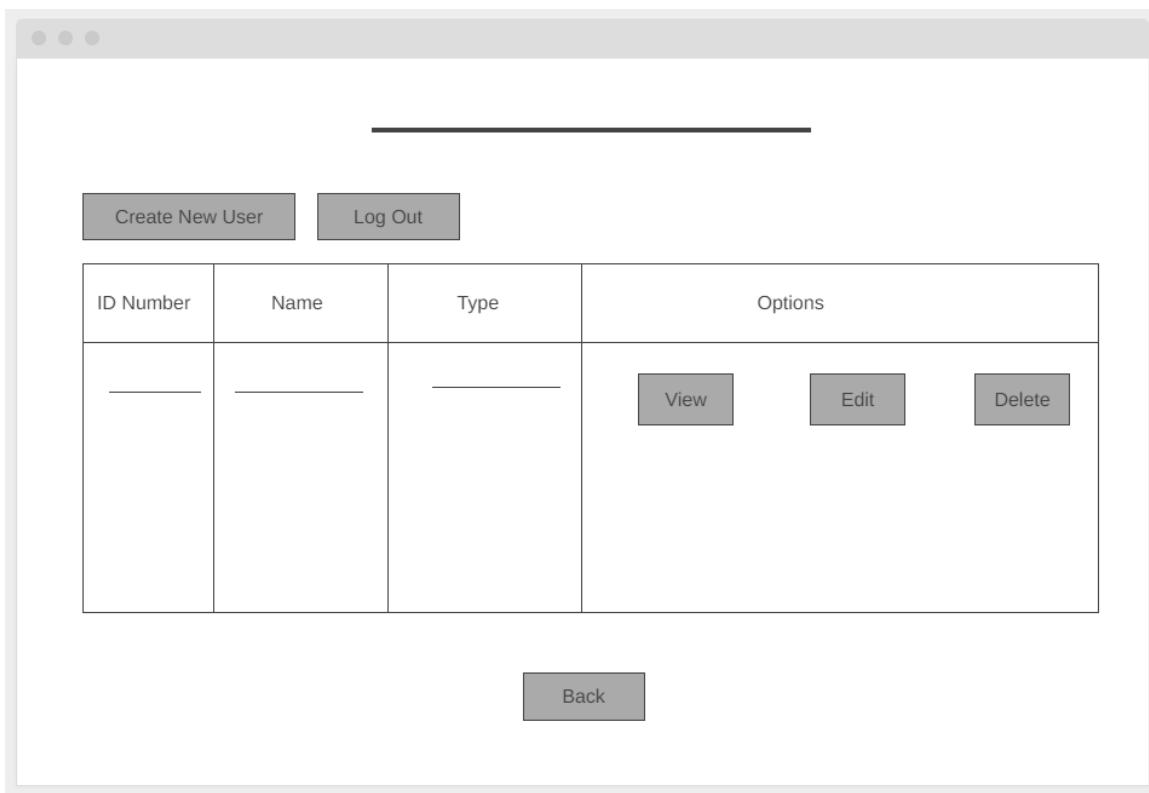
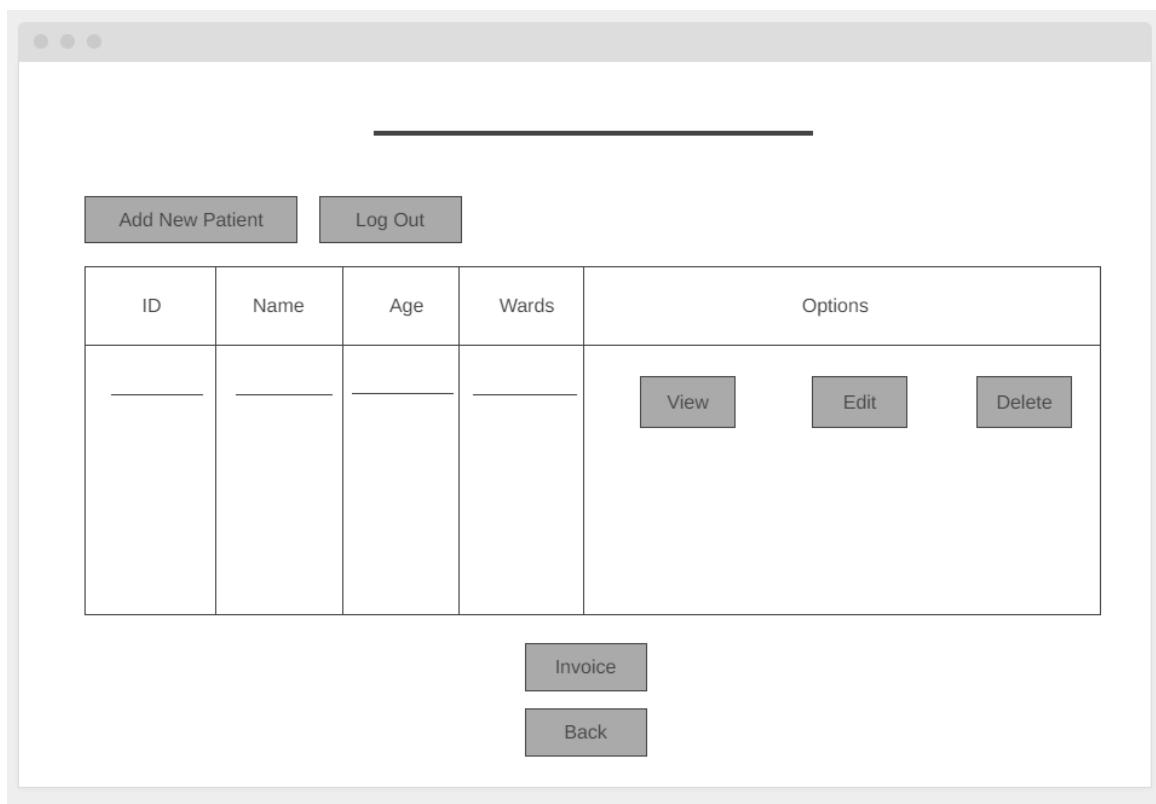


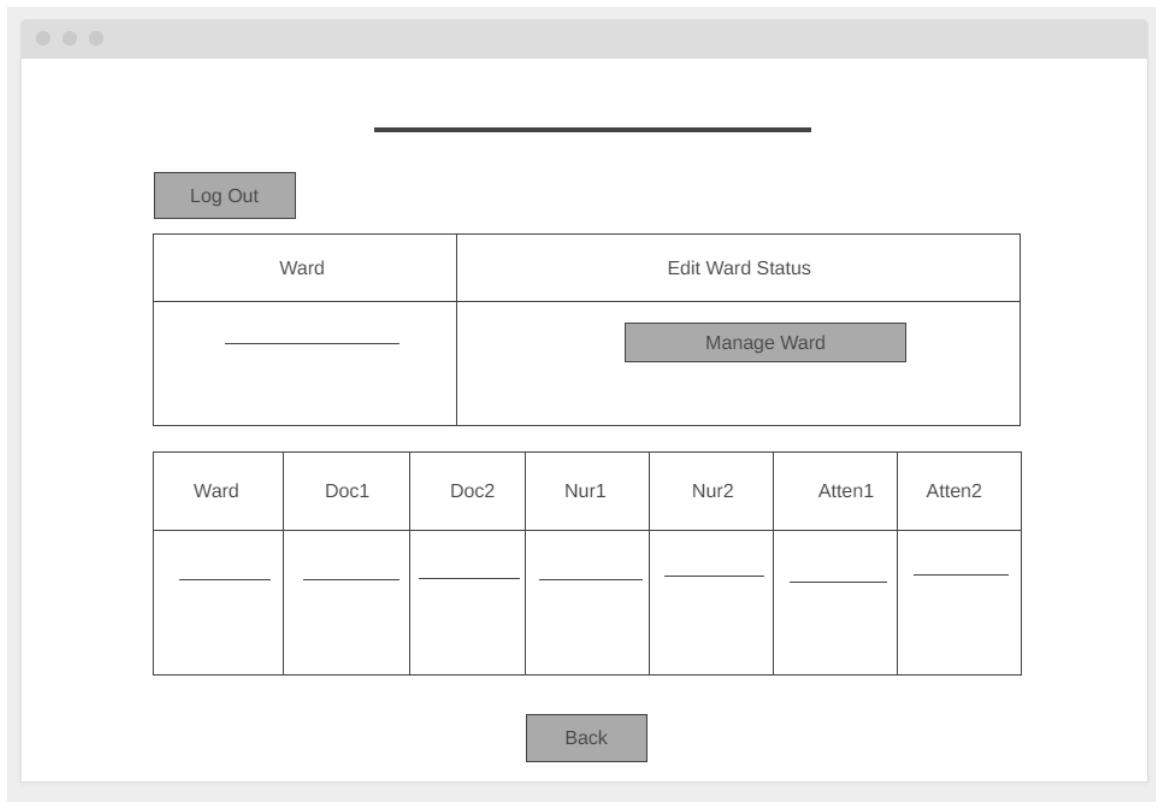
Figure 3. 29 Staff Page Wireframe



This wireframe shows the Patients Page. At the top, there are two buttons: "Add New Patient" and "Log Out". Below them is a table with columns: ID, Name, Age, Wards, and Options. The Options column contains three buttons: "View", "Edit", and "Delete". At the bottom of the page are two buttons: "Invoice" and "Back".

| ID | Name | Age | Wards | Options |
|-------|-------|-------|-------|----------------------------|
| _____ | _____ | _____ | _____ | View Edit Delete |

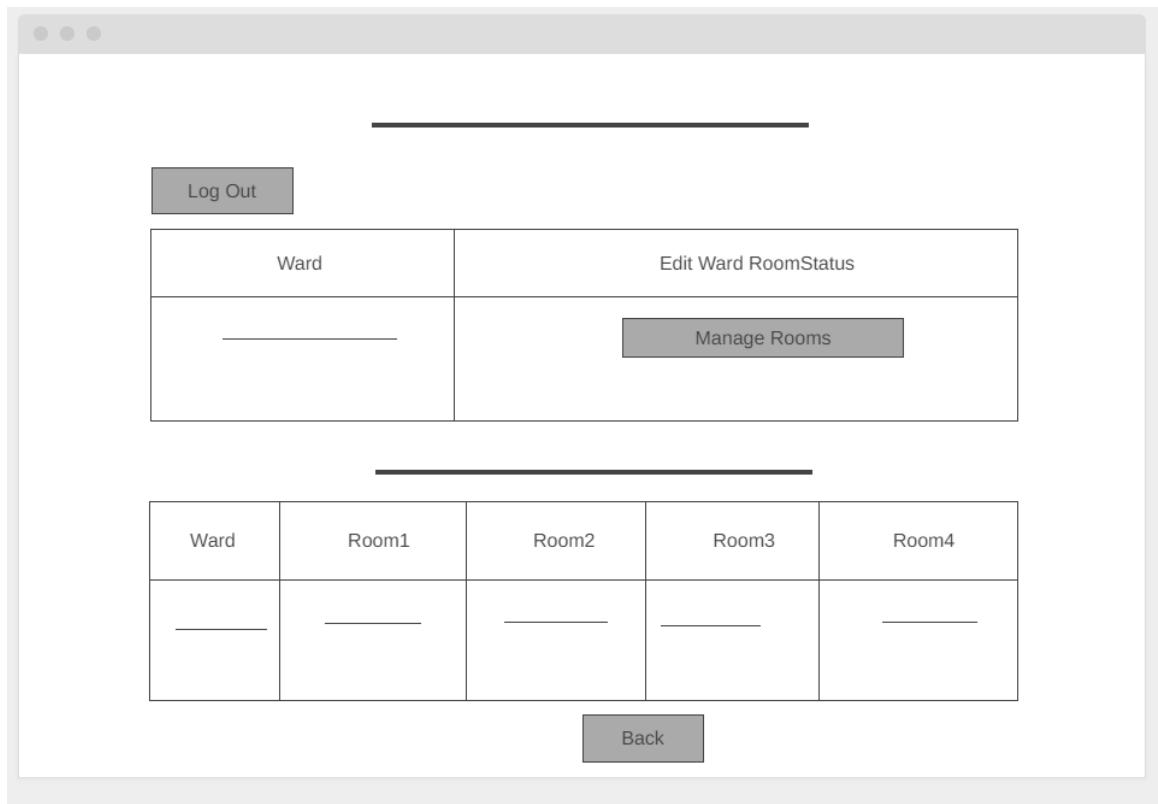
Figure 3. 30 Patients Page Wireframe



This wireframe shows the Wards Page. At the top, there is a "Log Out" button. Below it is a table with two columns: "Ward" and "Edit Ward Status". The "Edit Ward Status" column contains a "Manage Ward" button. At the bottom of the page is a table with seven columns: Ward, Doc1, Doc2, Nur1, Nur2, Atten1, and Atten2. At the very bottom is a "Back" button.

| Ward | Doc1 | Doc2 | Nur1 | Nur2 | Atten1 | Atten2 |
|-------|-------|-------|-------|-------|--------|--------|
| _____ | _____ | _____ | _____ | _____ | _____ | _____ |

Figure 3. 31 Wards Page Wireframe



The wireframe for the 'Rooms' page is organized into two main sections. The top section features a header with three dots on the left and a horizontal line in the center. Below this is a 'Log Out' button. A large rectangular area contains a 'Ward' label above a text input field, and an 'Edit Ward RoomStatus' label next to a 'Manage Rooms' button. Another horizontal line is positioned below this area. The bottom section contains a table with five columns labeled 'Ward', 'Room1', 'Room2', 'Room3', and 'Room4'. Each column has a corresponding text input field. At the bottom right of the table is a 'Back' button.

Figure 3. 32 Rooms Page Wireframe



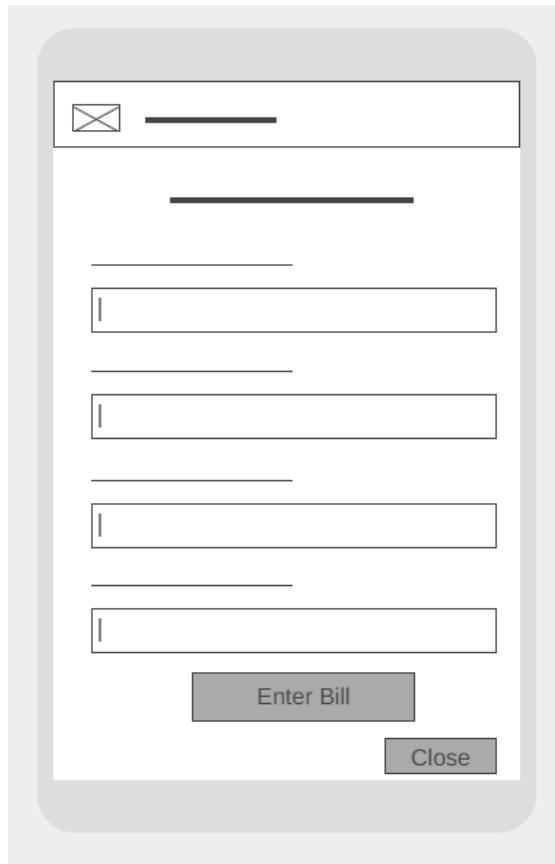
The wireframe for the 'Medicines' page includes a header with three dots on the left and a horizontal line in the center. It features two buttons: 'Add New Medicine' on the left and 'Log Out' on the right. Below these is a table with columns for 'ID', 'Name', 'Type', 'Quantity', and 'Options'. The 'Options' column contains three buttons: 'View', 'Edit', and 'Delete'. At the bottom of the page are two buttons: 'Invoice' and 'Back'.

Figure 3. 33 Medicines Page Wireframe



The wireframe shows a user interface for managing items. At the top, there are two buttons: "Add New Item" and "Log Out". Below this is a table with columns: ID, Name, Type, Quantity, and Options. The Options column contains three buttons: "View", "Edit", and "Delete". At the bottom of the page are two buttons: "Invoice" and "Back".

Figure 3. 34 Other Page Wireframe



The wireframe shows a modal dialog for entering a bill. It features a header with an envelope icon and a subject line. Below this are five input fields, each with a vertical cursor. At the bottom of the dialog are two buttons: "Enter Bill" and "Close".

Figure 3. 35 Patient Invoice Page Wireframe

3.3.2 Initial Output

Login Page :

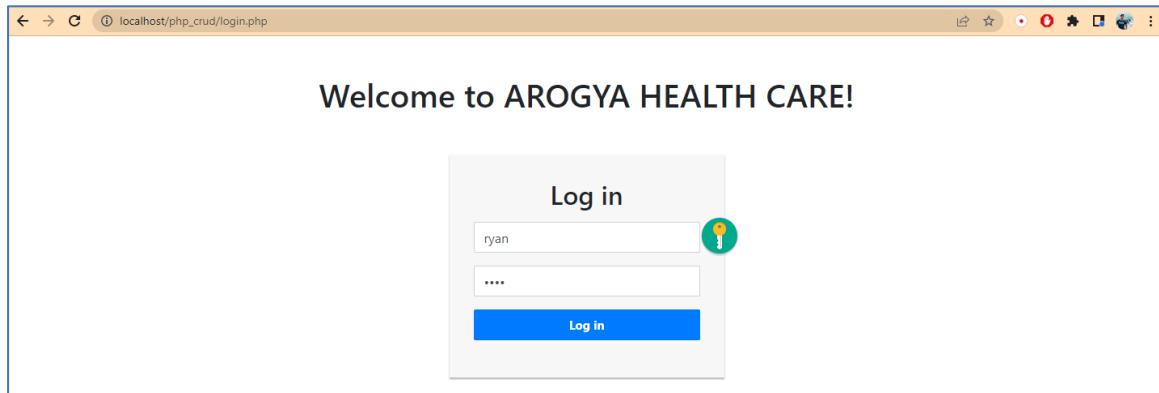


Figure 3. 36 Initial Output of Login Page

Home Page :

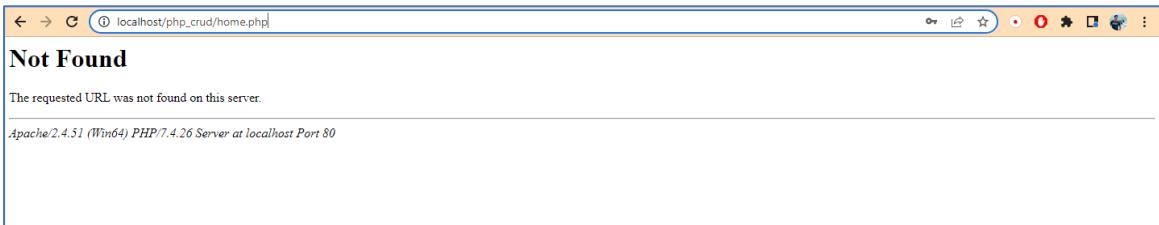


Figure 3. 37 Initial Output of Home Page

About Us Page :

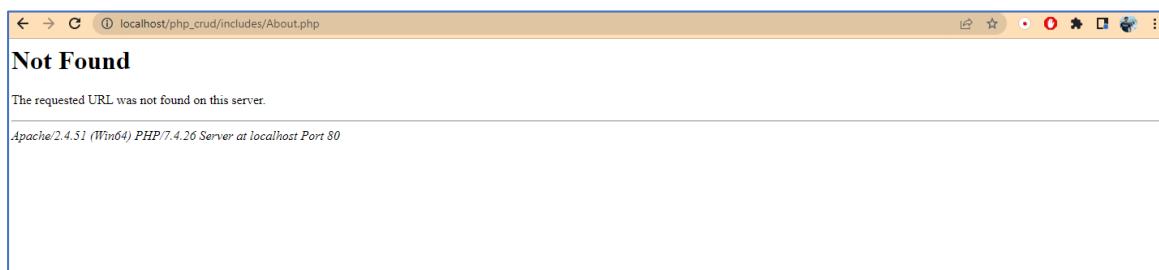


Figure 3. 38 Initial Output of About Us Page

Blog Page :

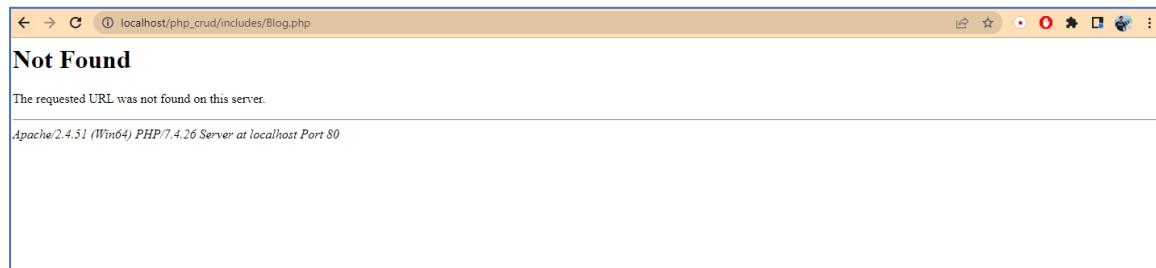


Figure 3. 39 Initial Output of Blog Page

Contact Us Page :



Figure 3. 40 Initial Output of Contact Us Page

Staff Page :

The screenshot shows a web browser window for the "AROGYA HEALTH CARE" system. The title bar indicates the page is "STAFF MANAGEMENT". The header includes links for "Home", "Patients", and "Schedule". Below the header is a navigation bar with "Create New User" and "Logout" buttons. The main content is a table titled "STAFF MANAGEMENT" with columns for "ID", "Name", "Type", and "CRUD Operations". The table lists seven staff members with their details and edit/delete buttons. At the bottom of the page is a "Back" button.

| ID | Name | Type | CRUD Operations | | |
|----|--------------------|------------|----------------------|----------------------|------------------------|
| 1 | Pavel Fernando | Doctor | View | EDIT | DELETE |
| 2 | Ryan Wickramaratne | Nurse | View | EDIT | DELETE |
| 3 | Judith Michelle | Technician | View | EDIT | DELETE |
| 4 | Kyle Wickramaratne | Attendant | View | EDIT | DELETE |
| 5 | Sadew Fernando | Doctor | View | EDIT | DELETE |
| 6 | Mashi Perera | Nurse | View | EDIT | DELETE |
| 7 | Shyni Perera | Attendant | View | EDIT | DELETE |

Figure 3. 41 Initial Output of Staff Page

Patients Page :

The screenshot shows a web browser window for 'AROGYA HEALTH CARE'. The title bar says 'localhost/php_crud/patients.php'. The main content area has a heading 'PATIENTS MANAGEMENT' and two buttons: 'Add New Patient' and 'Logout'. Below is a table with columns: ID, Name, Age, Ward, and CRUD Operations (View, Edit, Delete). The table contains four rows of patient data:

| ID | Name | Age | Ward | CRUD Operations |
|----|--------------------|-----|--------|------------------------|
| 1 | Judith Michelle | 20 | Ward 2 | View EDIT DELETE |
| 2 | Ryan Wickramaratne | 25 | Ward 1 | View EDIT DELETE |
| 3 | Pavel Fernando | 15 | Ward 3 | View EDIT DELETE |
| 4 | Malshan Fernando | 25 | Ward 2 | View EDIT DELETE |

A yellow 'Back' button is at the bottom left.

Figure 3. 42 Initial Output of Patients Page

Wards Page :

The screenshot shows a web browser window for 'AROGYA HEALTH CARE'. The title bar says 'localhost/php_crud/includes/wards.php'. The main content area has a heading 'WARDS' and a 'Logout' button. Below is a table with columns: WARD and STATUS. The table contains four rows of ward data:

| WARD | STATUS |
|--------|-----------------|
| Ward 1 | Ward Management |
| Ward 2 | Ward Management |
| Ward 3 | Ward Management |
| Ward 4 | Ward Management |

A yellow 'Back' button is at the bottom left.

Figure 3. 43 Initial Output of Wards Page

Rooms Page :

The screenshot shows a web browser window for 'localhost/php_crud/includesrooms.php'. The title bar says 'localhost/php_crud/includesrooms.php'. The main content area displays an error message: 'Not Found' followed by 'The requested URL was not found on this server.' and the server information 'Apache/2.4.51 (Win64) PHP/7.4.26 Server at localhost Port 80'.

Figure 3. 44 Initial Output of Rooms Page

Medicines Page :

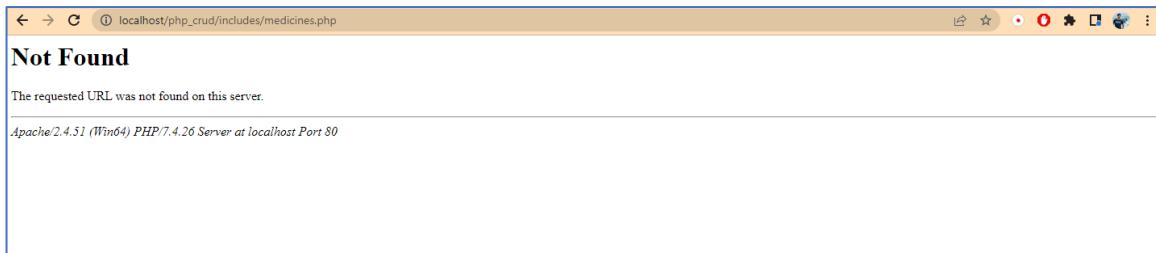


Figure 3. 45 Initial Output of Medicines Page

Other Page :

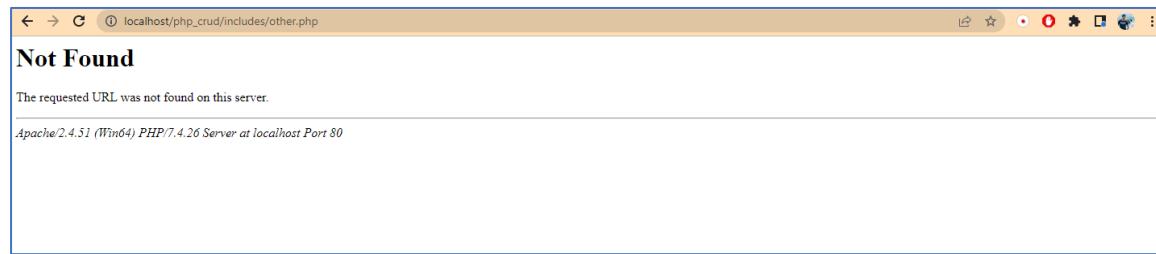


Figure 3. 46 Initial Output of Other Page

MySQL Database (“test” database) :

| Table | Action | Rows | Type | Collation | Size | Overhead |
|----------|---|------|--------|--------------------|----------|----------|
| admin | Browse Structure Search Insert Empty Drop | 3 | InnoDB | utf8mb4_general_ci | 16.0 Kib | - |
| patient | Browse Structure Search Insert Empty Drop | 1 | InnoDB | utf8mb4_general_ci | 16.0 Kib | - |
| staff | Browse Structure Search Insert Empty Drop | 6 | InnoDB | utf8mb4_general_ci | 16.0 Kib | - |
| wards | Browse Structure Search Insert Empty Drop | 4 | InnoDB | utf8mb4_general_ci | 16.0 Kib | - |
| 4 tables | Sum | 14 | MyISAM | utf8mb4_general_ci | 64.0 Kib | 0 B |

Figure 3. 47 MySQL Database (“test” database)

File Organization in WAMP Server :

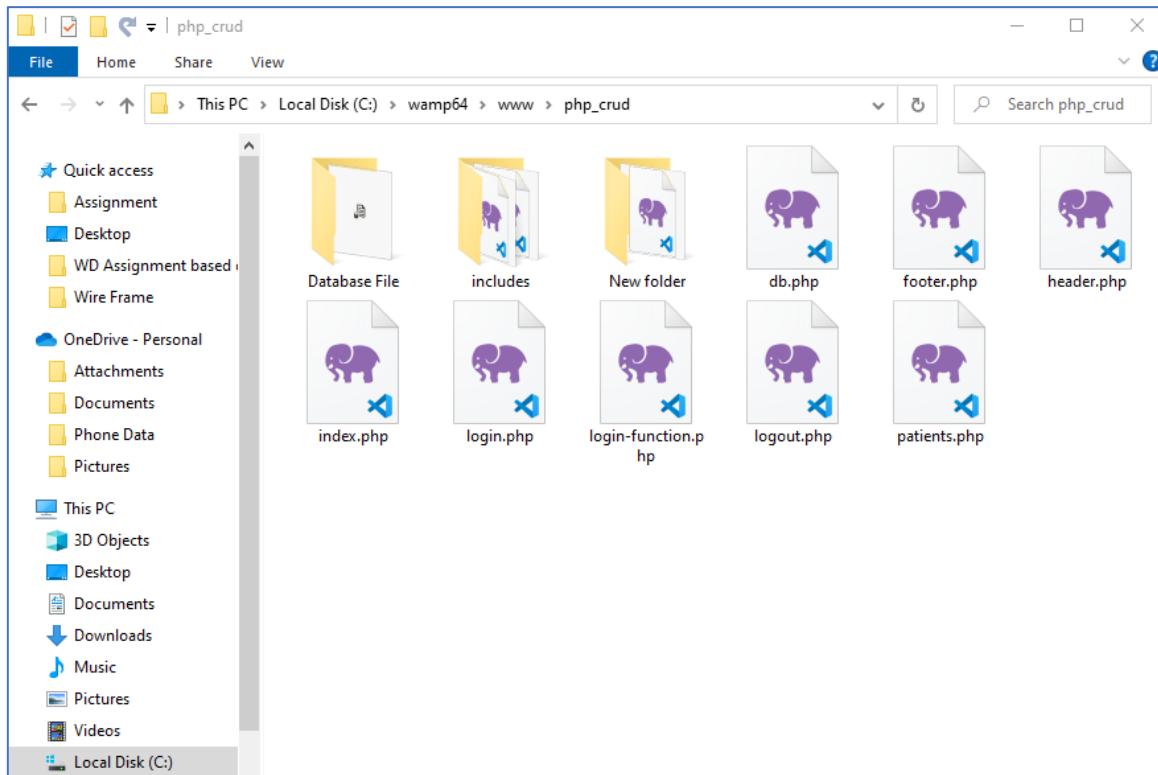


Figure 3. 48 File Organization in WAMP Server part 1

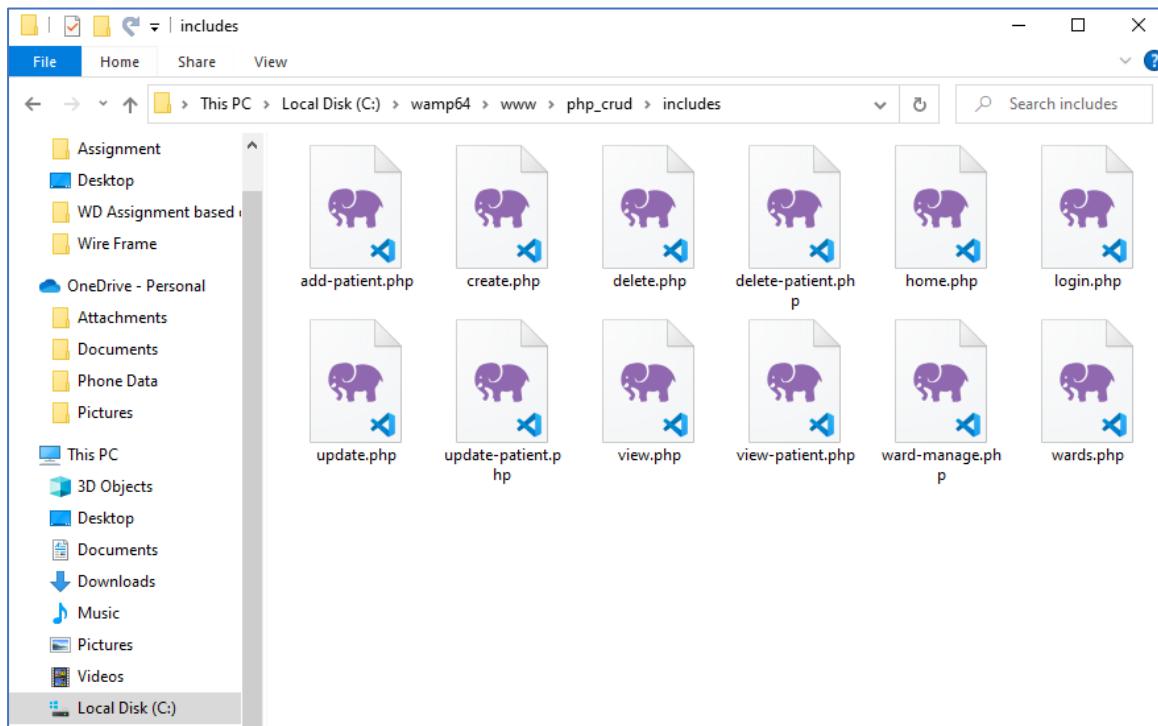


Figure 3. 49 File Organization in WAMP Server part 2

3.3.3 Final Design with Codes (Final Output)

Login Page

index.php :

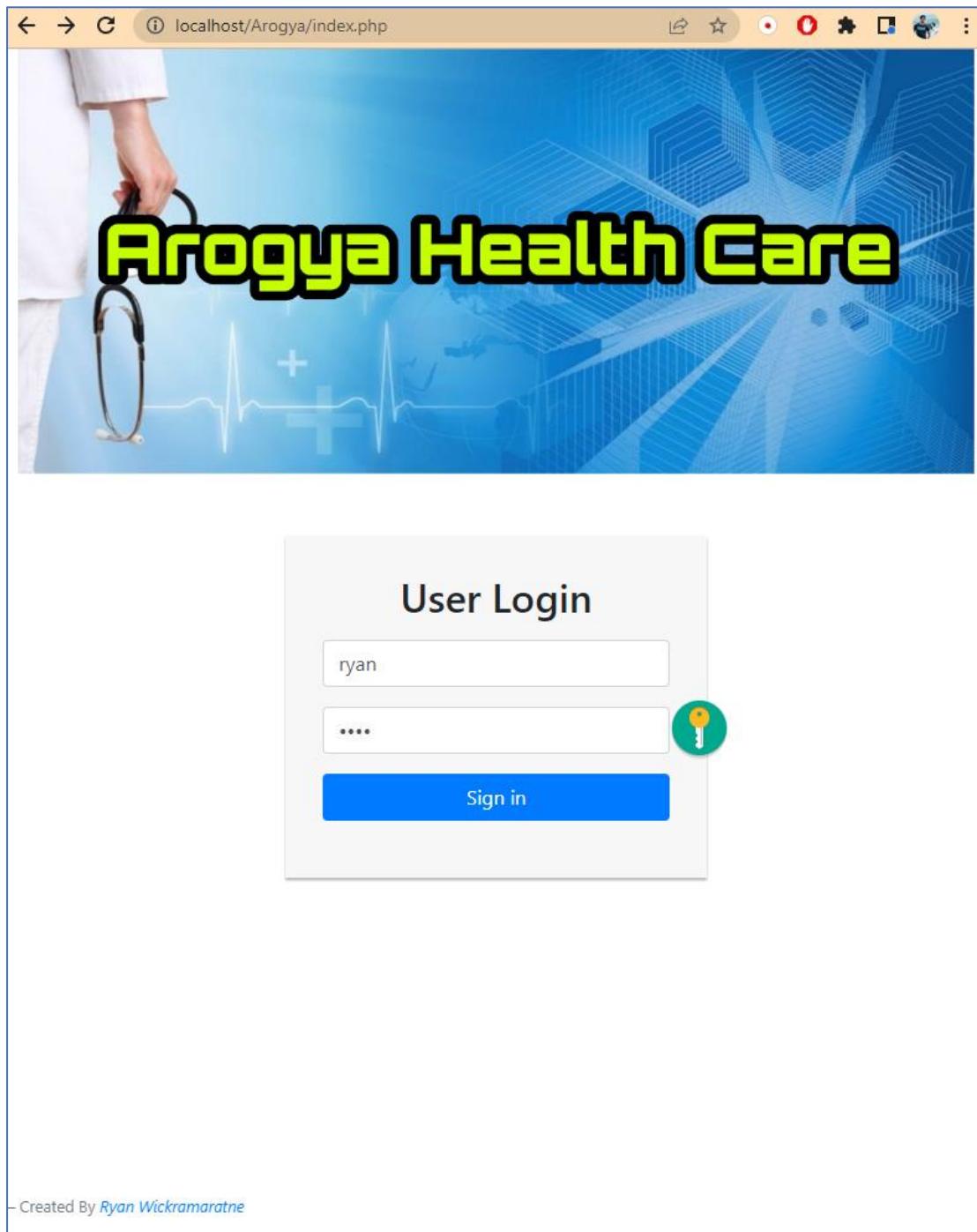
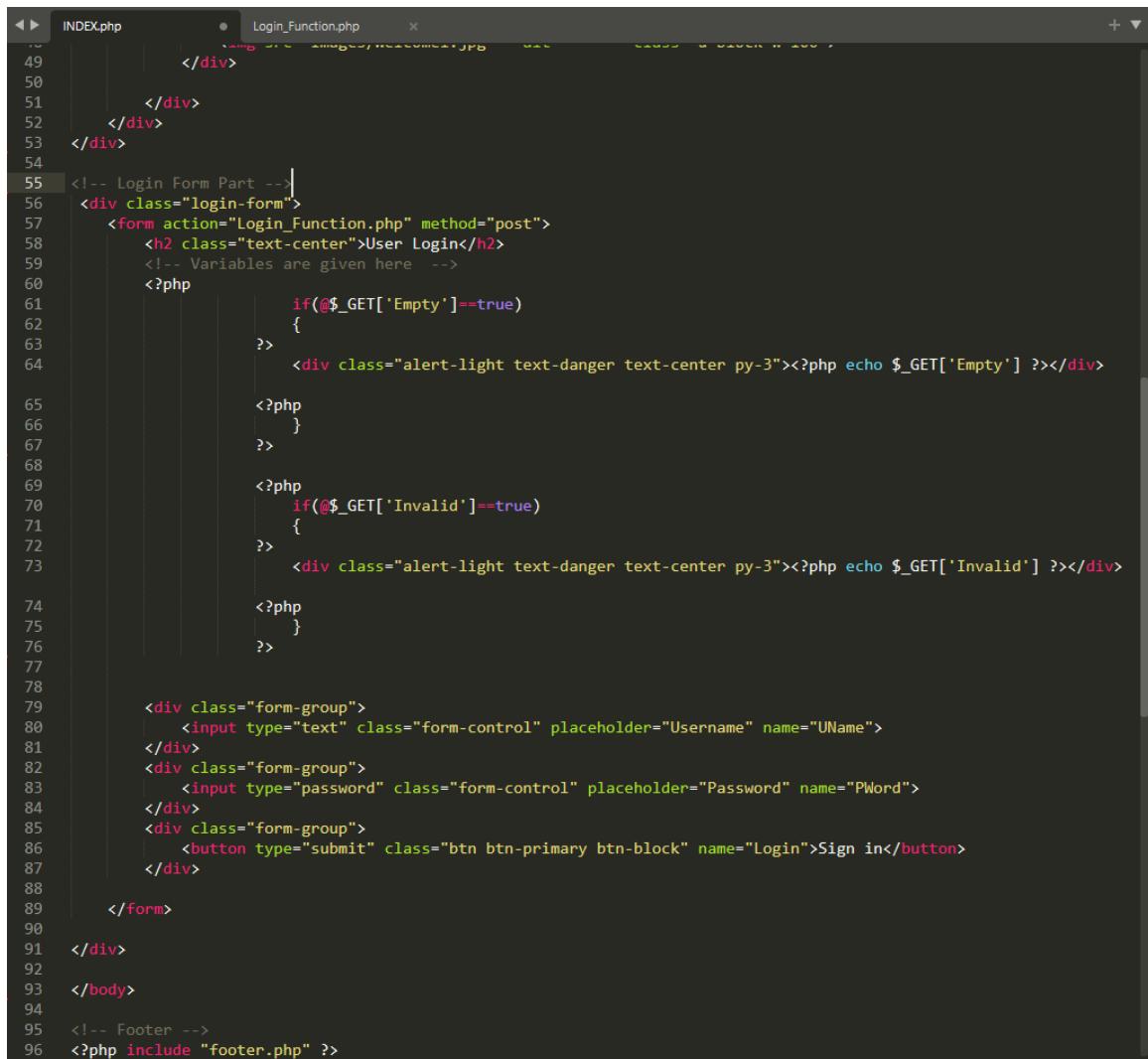


Figure 3. 50 Final Design of index.php

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7
8   <!-- Bootstrap CSS -->
9   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-eOJMYsd53ii+sc0/bJGfsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6" crossorigin="anonymous">
10
11   <!-- custom CSS File made by me -->
12   <link rel = "stylesheet" href="Mystyle.css">
13
14
15
16   <!-- Bootstrap CDN for icons -->
17   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.3.0/font/bootstrap-icons.css">
18
19   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">
20   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
21
22   <!-- Java Script CDN bundle -->
23   <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
24   <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"></script>
25   <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></script>
26
27
28   <title>Login Form</title>
29
30 </head>
31
32
33
34 <body>
35
36   <!-- Basic Carousel Part -->
37
38   <div class="container-fluid">
39     <div class="carousel slide" data-bs-ride="carousel">
40       <div class="carousel-inner">
41
42         <div class="carousel-item active">
43           
44         </div>
45
46         <div class="carousel-item">
47           
48         </div>
49
50   </div>
51 </div>
```

Figure 3. 51 Coding for index.php part 1

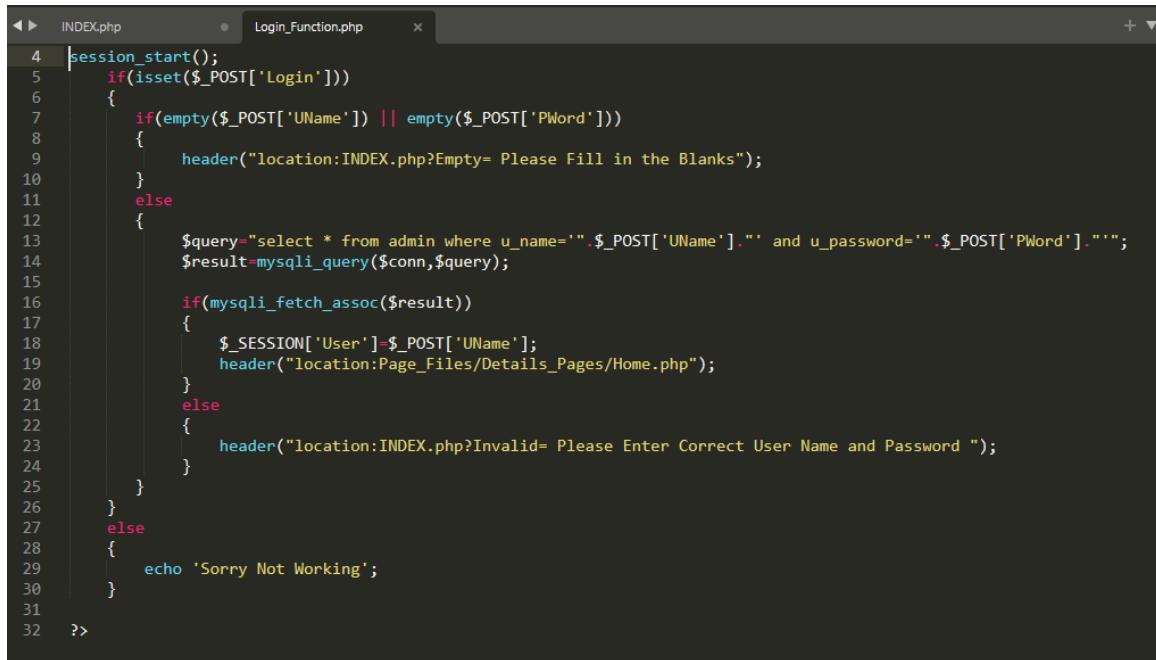


The screenshot shows a code editor window with two tabs: 'INDEX.php' and 'Login_Function.php'. The 'INDEX.php' tab is active, displaying the following PHP code:

```
49     </div>
50
51     </div>
52
53 </div>
54
55 <!-- Login Form Part -->
56 <div class="login-form">
57     <form action="Login_Function.php" method="post">
58         <h2 class="text-center">User Login</h2>
59         <!-- Variables are given here -->
60         <?php
61             if(@$_GET['Empty']==true)
62             {
63                 ?>          <div class="alert-light text-danger text-center py-3"><?php echo $_GET['Empty'] ?></div>
64
65                 ?>
66             }
67             ?>
68
69             <?php
70                 if(@$_GET['Invalid']==true)
71                 {
72                     ?>          <div class="alert-light text-danger text-center py-3"><?php echo $_GET['Invalid'] ?></div>
73
74                 <?php
75                 }
76             ?>
77
78         <div class="form-group">
79             <input type="text" class="form-control" placeholder="Username" name="UName">
80         </div>
81         <div class="form-group">
82             <input type="password" class="form-control" placeholder="Password" name="PWord">
83         </div>
84         <div class="form-group">
85             <button type="submit" class="btn btn-primary btn-block" name="Login">Sign in</button>
86         </div>
87
88     </form>
89
90 </div>
91
92 </body>
93
94 <!-- Footer -->
95 <?php include "footer.php" ?>
```

Figure 3. 52 Coding for index.php part 2

Login_Function.php :



```
INDEX.php      Login_Function.php      + ▾

4 | session_start();
5 | if(isset($_POST['Login']))
6 |
7 |     if(empty($_POST['UName']) || empty($_POST['PWord']))
8 |     {
9 |         header("location:INDEX.php?Empty= Please Fill in the Blanks");
10|     }
11| else
12|
13|     $query="select * from admin where u_name='".$_POST['UName']."' and u_password='".$_POST['PWord']."' ";
14|     $result=mysqli_query($conn,$query);
15|
16|     if(mysqli_fetch_assoc($result))
17|     {
18|         $_SESSION['User']=$_POST['UName'];
19|         header("location:Page_Files/Details_Pages/Home.php");
20|     }
21|     else
22|     {
23|         header("location:INDEX.php?Invalid= Please Enter Correct User Name and Password ");
24|     }
25| }
26| else
27| {
28|     echo 'Sorry Not Working';
29| }
30|
31|
32| ?>
```

Figure 3. 53 Coding for Login_Function.php

Home Page

Home.php :

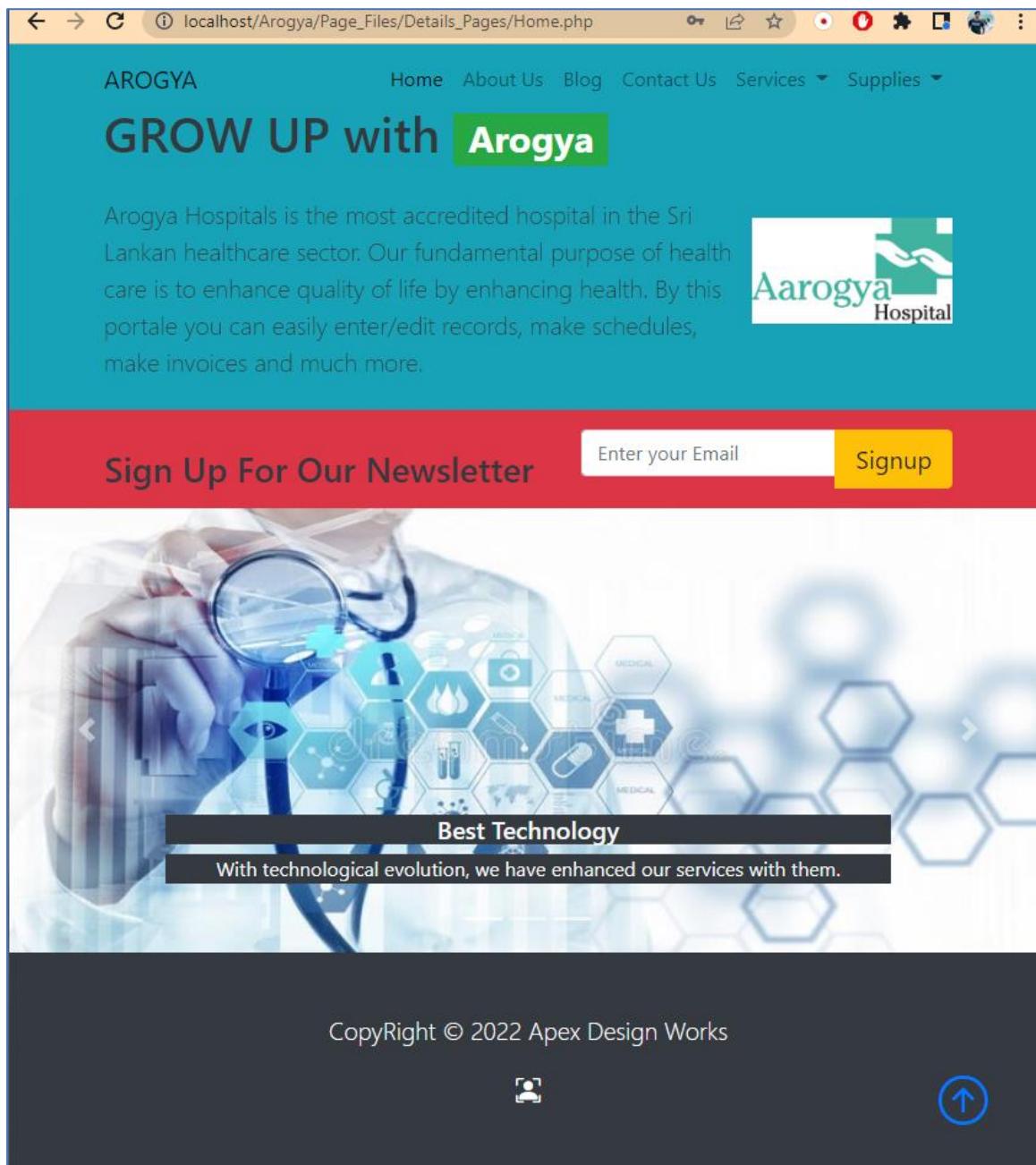


Figure 3. 54 Final Design of Home.php

```
1 <?php
2
3 session_start();
4 if(!isset($_SESSION['User'])){
5 header("location:../../INDEX.php");
6 }
7
8 ?>
9
10 <!-- Header -->
11 <?php include "../../header.php"?>
12
13 <!-- ----- Nav Bar
----->
14 <nav class="navbar navbar-light bg-info navbar-expand-sm fixed-top">
15
16     <div class="container">
17
18         <a href="#" class="navbar-brand" > AROGYA </a>
19
20         <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavDropdown">
21             <span class="navbar-toggler-icon"></span> </button>
22
23         <div class="collapse navbar-collapse" id="navbarNavDropdown" > ! -- Nav Bar can collapse
24 like a drop down button -->
25
26             <ul class="navbar-nav ms-auto">
27                 <li class="nav-item"> <a href="Home.php" class="nav-link active" > Home </a>
28                     </li>
29                 <li class="nav-item"> <a href="About.php" class="nav-link" > About Us </a>
30                     </li>
31                 <li class="nav-item"> <a href="Blog.php" class="nav-link" > Blog </a>
32                     </li>
33                 <li class="nav-item"> <a href="Contact.php" class="nav-link" > Contact Us </a>
34                     </li>
35
36                 <li class="nav-item dropdown">
37                     <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-toggle="dropdown" > Services </a>
38
39                     <div class="dropdown-menu">
40                         <a class="dropdown-item" href="../../Staff.php" >Staff</a>
41                         <a class="dropdown-item" href="../../Patients.php" >Patients</a>
42                         <a class="dropdown-item" href="../../Wards.php" >Wards</a>
43                         <a class="dropdown-item" href="../../Rooms.php" >Rooms</a>
44                     </div>
45                 </li>
46
47                 <li class="nav-item dropdown">
48                     <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-toggle="dropdown" > Supplies </a>
49                 </li>
50             </ul>
51         </div>
52     </div>
53 </nav>
```

Figure 3. 55 Coding for Home.php part 1

```
Home.php
41 ▼      <li class="nav-item dropdown">
42          <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-toggle="dropdown" > Supplies </a>
43
44          <div class="dropdown-menu">
45              <a class="dropdown-item" href="../Medicines.php">Medicines</a>
46              <a class="dropdown-item" href="../Other.php">Other</a>
47          </div>
48      </li>
49  </ul>
50
51      </div>
52  </div>
53
54 </nav>
55
56 <!-- Show Case
57 ----->
58
59 <section class="bg-info text-dark text-sm-start pt-5 pt-lg-5">
60     <div class="container">
61         <div class="d-md-flex align-items-center justify-content-between">
62
63             <div>
64                 <h1> GROW UP with <span class="badge bg-success">Arogya</span> </h1>
65                 <p class="lead my-4">Arogya Hospitals is the most accredited hospital in the Sri Lankan healthcare sector. Our fundamental purpose of health care is to enhance quality of life by enhancing health. By this portale you can easily enter/edit records, make schedules, make invoices and much more. </p>
66             </div>
67
68             <br>
69             
70         </div>
71
72     </div>
73 </div>
74 </section>
75
76 <!-- Signup for Newsletter Part
77 ----->
78
79 <section class="bg-danger text-dark pt-3">
80     <div class="container">
81         <div class="d-md-flex align-items-center justify-content-between">
82
83             <div>
84                 <h3 class="mb-3 mb-md-0"> Sign Up For Our Newsletter </h3>
85             </div>
86         </div>

```

Figure 3. 56 Coding for Home.php part 2

```
Home.php

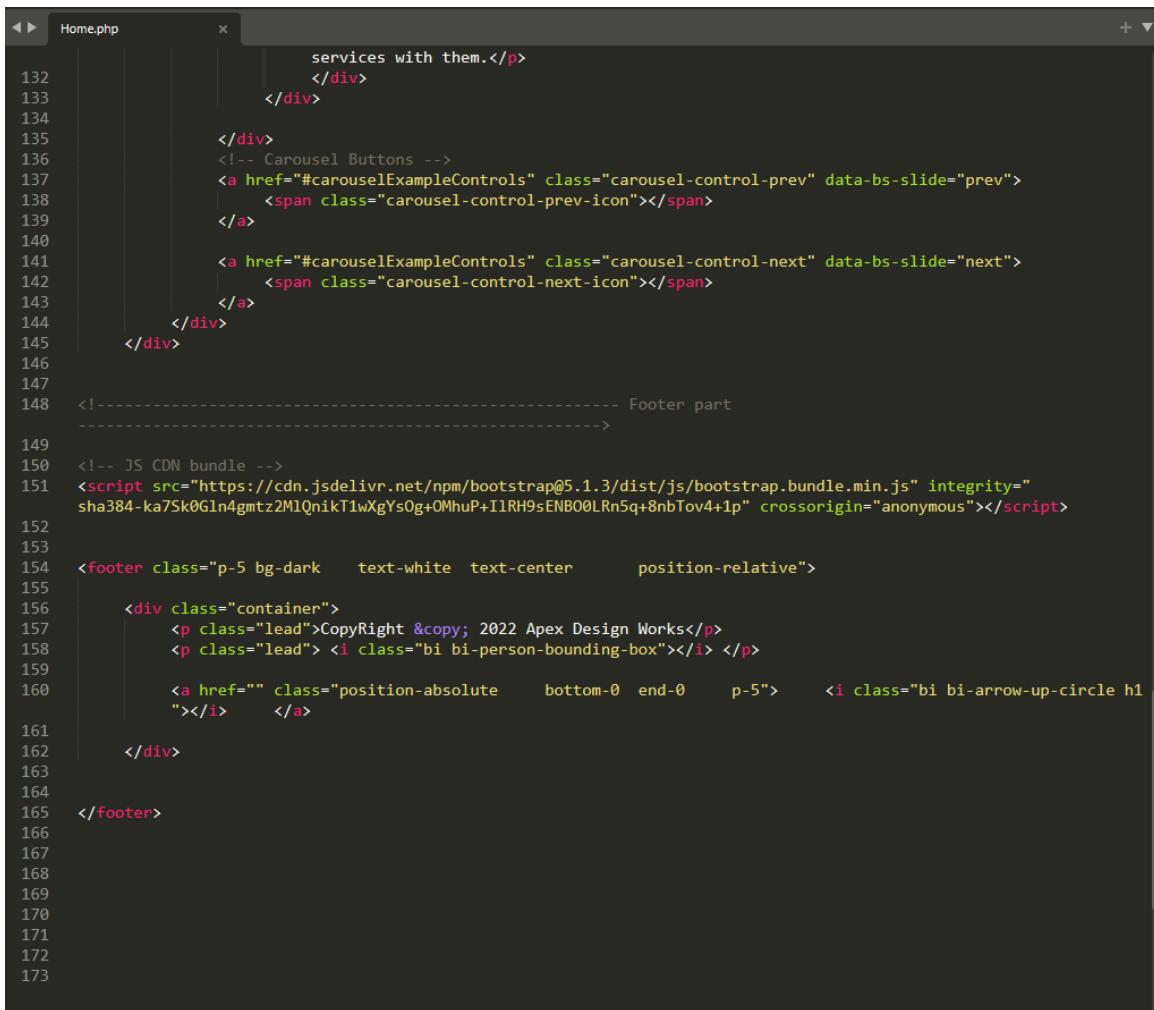
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131

    <div class="input-group mb-3 news-input">
        <input type="text" class="form-control" placeholder="Enter your Email">
        <button class="btn btn-warning btn-lg" type="button" id="button-addon2">Signup</button>
    </div>
</div>
</div>
</section>

<!-- Make Responsive Carousels --&gt;

&lt;div id="carouselExampleControls" class="carousel slide carousel-fade" data-bs-ride="carousel"&gt;
    &lt;div class="carousel-indicators"&gt;
        &lt;button type="button" data-bs-target="#carouselExampleControls" data-bs-slide-to="0" class="active" aria-current="true" aria-label="Slide 1"&gt;&lt;/button&gt;
        &lt;button type="button" data-bs-target="#carouselExampleControls" data-bs-slide-to="1" aria-label="Slide 2"&gt;&lt;/button&gt;
        &lt;button type="button" data-bs-target="#carouselExampleControls" data-bs-slide-to="2" aria-label="Slide 3"&gt;&lt;/button&gt;
    &lt;/div&gt;
    &lt;div class="carousel-inner"&gt;
        &lt;div class="carousel-item active" data-bs-interval="2000"&gt;
            &lt;img src="images/Hospital1.jpg" alt="" class="d-block w-100"&gt;
            &lt;div class="carousel-caption"&gt;
                &lt;h5 class="text-light bg-dark"&gt;Best Facilities&lt;/h5&gt;
                &lt;p class="text-light bg-dark"&gt;Arogya Hospital has the best equipment and facilities with the best service you will ever have.&lt;/p&gt;
            &lt;/div&gt;
        &lt;/div&gt;
        &lt;div class="carousel-item" data-bs-interval="1000"&gt;
            &lt;img src="images/Hospital2.jpg" alt="" class="d-block w-100"&gt;
            &lt;div class="carousel-caption"&gt;
                &lt;h5 class="text-light bg-dark"&gt;Best Treatment&lt;/h5&gt;
                &lt;p class="text-light bg-dark"&gt;We provide superior treatment for patients with superb friendly staff.&lt;/p&gt;
            &lt;/div&gt;
        &lt;/div&gt;
        &lt;div class="carousel-item" data-bs-interval="1000"&gt;
            &lt;img src="images/Hospital3.jpg" alt="" class="d-block w-100"&gt;
            &lt;div class="carousel-caption"&gt;
                &lt;h5 class="text-light bg-dark"&gt;Best Technology&lt;/h5&gt;
                &lt;p class="text-light bg-dark"&gt;With technological evolution, we have enhanced our services with them.&lt;/p&gt;
            &lt;/div&gt;
        &lt;/div&gt;
    &lt;/div&gt;
&lt;/div&gt;</pre>
```

Figure 3. 57 Coding for Home.php part 3



The screenshot shows a code editor window with the file 'Home.php' open. The code is a continuation of a previous part, starting with line 148. It includes a footer section with a copyright notice and a navigation link, followed by a script tag for a JS CDN bundle. The code uses Bootstrap classes like 'p-5', 'bg-dark', 'text-white', 'text-center', and 'position-relative'. It also includes icons from the Bootstrap library.

```
132     services with them.</p>
133     </div>
134
135     </div>
136     <!-- Carousel Buttons -->
137     <a href="#carouselExampleControls" class="carousel-control-prev" data-bs-slide="prev">
138         <span class="carousel-control-prev-icon"></span>
139     </a>
140
141     <a href="#carouselExampleControls" class="carousel-control-next" data-bs-slide="next">
142         <span class="carousel-control-next-icon"></span>
143     </a>
144
145 </div>
146
147
148 <!-- Footer part
----->
149 <!-- JS CDN bundle -->
150 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMuP+IlRH9sENB00LRn5q+8nbTov4+lP" crossorigin="anonymous"></script>
151
152
153 <footer class="p-5 bg-dark    text-white    text-center      position-relative">
154
155     <div class="container">
156         <p class="lead">CopyRight © 2022 Apex Design Works</p>
157         <p class="lead"> <i class="bi bi-person-bounding-box"></i> </p>
158
159         <a href="" class="position-absolute    bottom-0    end-0    p-5" style="color: white; text-decoration: none; font-size: 2em; opacity: 0.5;">
160             <i class="bi bi-arrow-up-circle h1" style="font-size: 1.5em; color: white; opacity: 0.5;"></i>
161         </a>
162     </div>
163
164
165 </footer>
166
167
168
169
170
171
172
173
```

Figure 3. 58 Coding for Home.php part 4

About Us Page

About.php :

The screenshot shows a web browser window with the URL localhost/Arogya/Page_Files/Details_Pages/About.php. The page has a teal header bar with the logo 'AROGYA' and navigation links: Home, About Us, Blog, Contact Us, Services, and Supplies. Below the header are three dark grey boxes with white icons and text:

- Awarded**: Features a trophy icon. Text: "We have been recognized for our exceptional service and care by both international and local awarding bodies." A blue 'Read More>' button is at the bottom.
- Secured**: Features a shield with a checkmark icon. Text: "Your health is our priority. We give our priority to helping you, healing you, and protecting you." A blue 'Read More>' button is at the bottom.
- Excellent Staff**: Features a people icon. Text: "We have an expert medical staff to provide the best health recovery possible. So don't worry about your health." A blue 'Read More>' button is at the bottom.

Below these cards is a teal section titled **Our Specialists**. Text: "Our Specialists all have remarkable working experience with international recognition." Two circular profile pictures are shown: one of a man labeled **Prof. Shaman** and one of a woman labeled **Prof. Jithangi**.

Figure 3. 59 Final Design of About.php part 1

The screenshot shows a web browser displaying the 'About Us' section of the AROGYA website. The top navigation bar includes links for Home, About Us, Blog, Contact Us, Services, and Supplies. Below the navigation, there are two profile boxes for medical professionals:

- Prof. Pujitha Wickramasinghe**:
Training in Paediatric Gastroenterology and Clinical Nutrition Royal Children's Hospital, and Research on Human Body Composition at Children's Nutrition Research Centre, Dept of Paediatrics and Child Health Brisbane, Australia (Clinical Nutrition Research Fellow, Post MD overseas training) – Apr 2003 – Apr 2004.
Social media icons for Twitter, Facebook, LinkedIn, and Instagram are present below her photo.
- Prof. Shamya De Silva**:
Kirikankanage Shamya Harshini de Silva joined the Faculty of Medicine, University of Colombo as a Senior Lecturer in September 1991 and was subsequently promoted to the current position of Professor in Paediatrics in November 2009. She is a honorary Consultant Paediatrician attached to the University Unit at Lady Ridgeway Hospital since September 1991.
Social media icons for Twitter, Facebook, LinkedIn, and Instagram are present below her photo.

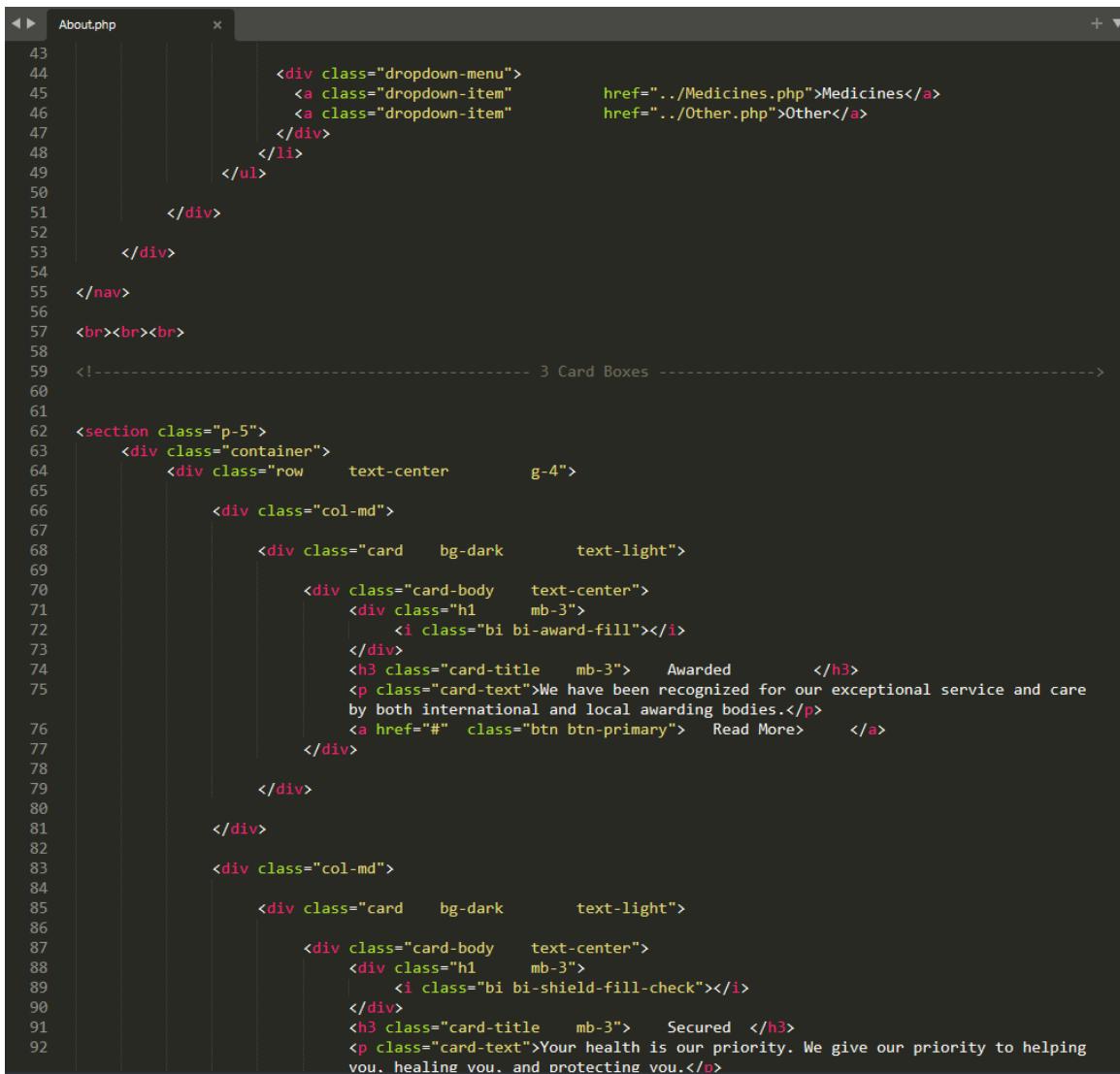
At the bottom of the page, a dark footer bar contains the text "CopyRight © 2022 Apex Design Works" and two small circular icons: one with a person icon and another with an upward arrow icon.

Figure 3. 60 Final Design of About.php part 2

```

1 <?php
2
3 session_start();
4 if(!isset($_SESSION['User'])){
5 header("location:../../INDEX.php");
6 }
7
8 ?>
9
10 <!-- Header -->
11 <?php include "../../header.php"?>
12
13 <!-- ----- Nav Bar
14 <nav class="navbar      navbar-light      bg-info      navbar-expand-sm      fixed-top">
15
16   <div class="container">
17
18     <a href="#" class="navbar-brand" > AROGYA </a>
19
20     <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavDropdown"> <
21       span class="navbar-toggler-icon"></span> </button>
22
23     <div class="collapse      navbar-collapse" id="navbarNavDropdown" > <! -- Nav Bar can collapse
24       like a drop down button -->
25
26       <ul class="navbar-nav      ms-auto">
27         <li class="nav-item"> <a href="Home.php" class="nav-link" > Home </a>
28         </li>
29         <li class="nav-item"> <a href="About.php" class="nav-link active" > About Us </a>
30         </li>
31         <li class="nav-item"> <a href="Blog.php" class="nav-link" > Blog </a>
32         </li>
33         <li class="nav-item"> <a href="Contact.php" class="nav-link" > Contact Us </a>
34         </li>
35
36         <li class="nav-item dropdown">
37           <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-toggle="dropdown" > Services </a>
38
39           <div class="dropdown-menu">
40             <a class="dropdown-item" href="../Staff.php">Staff</a>
41             <a class="dropdown-item" href="../Patients.php">Patients</a>
42             <a class="dropdown-item" href="../Wards.php">Wards</a>
43             <a class="dropdown-item" href="../Rooms.php">Rooms</a>
44           </div>
45         </li>
46
47         <li class="nav-item dropdown">
48           <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-toggle="dropdown" > Supplies </a>
49         </li>
50       </ul>
51     </div>
52   </div>
53 </nav>
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
778
779
779
780
781
782
783
784
785
786
787
788
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
887
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
917
918
919
919
920
921
922
923
924
925
926
927
928
928
929
929
930
931
932
933
934
935
936
937
938
938
939
939
940
941
942
943
944
945
946
947
947
948
949
949
950
951
952
953
954
955
956
957
958
958
959
959
960
961
962
963
964
965
966
967
967
968
969
969
970
971
972
973
974
975
976
977
977
978
978
979
979
980
981
982
983
984
985
986
986
987
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1027
1028
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1057
1058
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1071
1072
1073
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1081
1082
1083
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1091
1092
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1101
1102
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1111
1112
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1121
1122
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1131
1132
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1141
1142
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1151
1152
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1161
1162
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1171
1172
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1181
1182
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1191
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1201
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1211
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1221
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1231
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1241
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1251
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1261
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1271
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1281
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1291
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1301
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1311
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1321
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1331
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1341
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1351
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1361
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1371
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1381
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1391
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1401
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1411
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1421
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1431
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1441
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1451
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1461
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1471
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1481
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1491
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1501
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1511
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1521
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1531
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1541
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1551
1552
1553
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1561
1562
1563
1563
1564
1564
1565
1565
1566
1566
1567
1567
1568
1568
1569
1569
1570
1571
1572
1573
1573
1574
1574
1575
1575
1576
1576
1577
1577
1578
1578
1579
1579
1580
1581
1582
1583
1583
1584
1584
1585
1585
1586
1586
1587
1587
1588
1588
1589
1589
1590
1591
1592
1593
1593
1594
1594
1595
1595
1596
1596
1597
1597
1598
1598
1599
1599
1600
1601
1602
1603
1603
1604
1604
1605
1605
1606
1606
1607
1607
1608
1608
1609
1609
1610
1611
1612
1613
1613
1614
1614
1615
1615
1616
1616
1617
1617
1618
1618
1619
1619
1620
1621
1622
1623
1623
1624
1624
1625
1625
1626
1626
1627
1627
1628
1628
1629
1629
1630
1631
1632
1633
1633
1634
1634
1635
1635
1636
1636
1637
1637
1638
1638
1639
1639
1640
1641
1642
1643
1643
1644
1644
1645
1645
1646
1646
1647
1647
1648
1648
1649
1649
1650
1651
1652
1653
1653
1654
1654
1655
1655
1656
1656
1657
1657
1658
1658
1659
1659
1660
1661
1662
1663
1663
1664
1664
1665
1665
1666
1666
1667
1667
1668
1668
1669
1669
1670
1671
1672
1673
1673
1674
1674
1675
1675
1676
1676
1677
1677
1678
1678
1679
1679
1680
1681
1682
1683
1683
1684
1684
1685
1685
1686
1686
1687
1687
1688
1688
1689
1689
1690
1691
1692
1693
1693
1694
1694
1695
1695
1696
1696
1697
1697
1698
1698
1699
1699
1700
1701
1702
1703
1703
1704
1704
1705
1705
1706
1706
1707
1707
1708
1708
1709
1709
1710
1711
1712
1713
1713
1714
1714
1715
1715
1716
1716
1717
1717
1718
1718
1719
1719
1720
1721
1722
1723
1723
1724
1724
1725
1725
1726
1726
1727
1727
1728
1728
1729
1729
1730
1731
1732
1733
1733
1734
1734
1735
1735
1736
1736
1737
1737
1738
1738
1739
1739
1740
1741
1742
1743
1743
1744
1744
1745
1745
1746
1746
1747
1747
1748
1748
1749
1749
1750
1751

```



The screenshot shows a code editor window titled "About.php". The code is written in PHP and HTML. It includes a navigation bar with a dropdown menu, three card boxes, and some descriptive text. The code uses Bootstrap classes like "dropdown-menu", "card", "text-center", and "bg-dark". It also includes icons from the Bootstrap library.

```
43     <div class="dropdown-menu">
44         <a class="dropdown-item" href="../Medicines.php">Medicines</a>
45         <a class="dropdown-item" href="../Other.php">Other</a>
46     </div>
47     </li>
48 </ul>
49 </div>
50 </div>
51 </div>
52 <br><br><br>
53 <br><br><br>
54 <!-- ----- 3 Card Boxes ----->
55 <!-- ----- 3 Card Boxes ----->
56 <!-- ----- 3 Card Boxes ----->
57 <!-- ----- 3 Card Boxes ----->
58 <!-- ----- 3 Card Boxes ----->
59 <!-- ----- 3 Card Boxes ----->
60 <!-- ----- 3 Card Boxes ----->
61 <!-- ----- 3 Card Boxes ----->
62 <section class="p-5">
63     <div class="container">
64         <div class="row text-center g-4">
65             <div class="col-md">
66                 <div class="card bg-dark text-light">
67                     <div class="card-body text-center">
68                         <div class="h1 mb-3">
69                             <i class="bi bi-award-fill"></i>
70                         </div>
71                         <h3 class="card-title mb-3"> Awarded </h3>
72                         <p class="card-text">We have been recognized for our exceptional service and care by both international and local awarding bodies.</p>
73                         <a href="#" class="btn btn-primary"> Read More </a>
74                     </div>
75                 </div>
76             </div>
77         </div>
78     </div>
79     <div class="col-md">
80         <div class="card bg-dark text-light">
81             <div class="card-body text-center">
82                 <div class="h1 mb-3">
83                     <i class="bi bi-shield-fill-check"></i>
84                 </div>
85                 <h3 class="card-title mb-3"> Secured </h3>
86                 <p class="card-text">Your health is our priority. We give our priority to helping you, healing you, and protecting you.</p>
87             </div>
88         </div>
89     </div>
90 </div>
91 </div>
92 </div>
```

Figure 3. 62 Coding for About.php part 2

```

About.php

you, healing you, and protecting you.</p>
<a href="#" class="btn btn-primary"> Read More </a>
</div>
</div>
</div>
</div>
<div class="col-md">
<div class="card bg-dark text-light">
<div class="card-body text-center">
<div class="h1 mb-3">
| <i class="bi bi-people-fill"></i>
</div>
<h3 class="card-title mb-3"> Excellent Staff </h3>
<p class="card-text">We have an expert medical staff to provide the best health recovery possible. So don't worry about your health.</p>
<a href="#" class="btn btn-primary"> Read More </a>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<!-- Our Instructors part
-->
<section id="instructors" class="p-5 bg-info">
<div class="container">
<h2 class="text-center text-dark">Our Specialists</h2>
<p class="lead text-center text-dark mb-5">Our Specialists all have remarkable working experience with international recognition.</p>
<div class="row g-4">
<div class="col-md-6 col-lg-3">
<div class="card bg-light">
<div class="card-body text-center">

<h3 class="card-title mb-3">Prof. Shaman Rajindrajith</h3>
<p class="card-text">I am a consultant Paediatrician with special interest to gastroenterology. I have experience in treating children with functional gastrointestinal disorders (constipation, abdominal pain etc..), inflammatory bowel disorders, and liver diseases. I also perform endoscopic procedures in children.</p>

```

Figure 3.63 Coding for About.php part 3

```

136             disorders, and liver diseases. I also perform endoscopic procedures in children.</p>
137             <a href=""><i class="bi bi-twitter text-dark mx-1"></i></a>
138             <a href=""><i class="bi bi-facebook text-dark mx-1"></i></a>
139             <a href=""><i class="bi bi-linkedin text-dark mx-1"></i></a>
140             <a href=""><i class="bi bi-instagram text-dark mx-1"></i></a>
141         </div>
142     </div>
143 </div>
144
145     <div class="col-md-6 col-lg-3">
146         <div class="card bg-light">
147             <div class="card-body text-center">
148                 
149                 <h3 class="card-title mb-3">Prof. Jithangi Wanigasinghe</h3>
150                 <p class="card-text">Prof. Jithangi Wanigasinghe is a professor in paediatric neurology in the Department of Paediatrics, Faculty of Medicine, University of Colombo. She graduated from the University of Colombo in 1998 with first Class Honours. Her main field of work relates to paediatric epilepsies and her research interests have been most on treatment of infantile spasms.</p>
151                 <a href=""><i class="bi bi-twitter text-dark mx-1"></i></a>
152                 <a href=""><i class="bi bi-facebook text-dark mx-1"></i></a>
153                 <a href=""><i class="bi bi-linkedin text-dark mx-1"></i></a>
154                 <a href=""><i class="bi bi-instagram text-dark mx-1"></i></a>
155             </div>
156         </div>
157     </div>
158
159     <div class="col-md-6 col-lg-3">
160         <div class="card bg-light">
161             <div class="card-body text-center">
162                 
163                 <h3 class="card-title mb-3">Prof. Pujitha Wickramasinghe</h3>
164                 <p class="card-text">Training in Paediatric Gastroenterology and Clinical Nutrition Royal Children's Hospital, and Research on Human Body Composition at Children's Nutrition Research Centre, Dept of Paediatrics and Child Health Brisbane, Australia (Clinical Nutrition Research Fellow, Post MD overseas training) - Apr 2003 - Apr 2004.</p>
165                 <a href=""><i class="bi bi-twitter text-dark mx-1"></i></a>
166                 <a href=""><i class="bi bi-facebook text-dark mx-1"></i></a>
167                 <a href=""><i class="bi bi-linkedin text-dark mx-1"></i></a>
168                 <a href=""><i class="bi bi-instagram text-dark mx-1"></i></a>
169             </div>
170         </div>
171     </div>
172
173     <div class="col-md-6 col-lg-3">
174         <div class="card bg-light">
175             <div class="card-body text-center">
176                 

```

Figure 3. 64 Coding for About.php part 4

```
174     <div class="card bg-light">
175         <div class="card-body text-center">
176             
177             <h3 class="card-title mb-3">Prof. Shamya De Silva</h3>
178             <p class="card-text">Kirikankanage Shamya Harshini de Silva joined the Faculty of Medicine, University of Colombo as a Senior Lecturer in September 1991 and was subsequently promoted to the current position of Professor in Paediatrics in November 2009. She is a honorary Consultant Paediatrician attached to the University Unit at Lady Ridgeway Hospital since September 1991.</p>
179             <a href="#"><i class="bi bi-twitter text-dark mx-1"></i></a>
180             <a href="#"><i class="bi bi-facebook text-dark mx-1"></i></a>
181             <a href="#"><i class="bi bi-linkedin text-dark mx-1"></i></a>
182             <a href="#"><i class="bi bi-instagram text-dark mx-1"></i></a>
183         </div>
184     </div>
185 </div>
186 </div>
187 </div>
188 </div>
189 </section>
190
191
192
193
194
195
196 <!------- Footer part ----->
197
198 <footer class="p-5 bg-dark text-white text-center position-relative">
199     <div class="container">
200         <p class="lead">CopyRight © 2022 Apex Design Works</p>
201         <p class="lead"> <i class="bi bi-person-bounding-box"></i> </p>
202
203         <a href="#" class="position-absolute bottom-0 end-0 p-5" style="color: inherit; text-decoration: none;">
204             <i class="bi bi-arrow-up-circle h1"></i>
205         </a>
206     </div>
207
208
209 </footer>
210
211
212
213
214
```

Figure 3. 65 Coding for About.php part 5

Blog Page

Blog.php :



The screenshot shows a web browser window displaying a blog post. The header bar includes back, forward, and search icons, along with the URL 'localhost/Arogya/Page_Files/Details_Pages/Blog.php'. The top navigation bar is teal and contains the logo 'AROGYA' and links for Home, About Us, Blog, Contact Us, Services, and Supplies. The main content area features a large title 'COVID-19 Treatment' in bold black font. Below the title is a paragraph of text: 'We offer level 1 care in four luxury hotels in the vicinity of Lanka Hospitals for asymptomatic & mildly symptomatic patients, with 24/7 hour medical & nursing care under the supervision of specialised consultants with advanced High Dependency Care.' To the left of the text is a photograph of a doctor in a white coat and mask examining a patient's arm. At the bottom right of the content area is a button labeled '> Read More'.

Figure 3. 66 Final Design of Blog.php part 1

The screenshot shows a web browser window with the URL `localhost/Arogya/Page_Files/Details_Pages/Blog.php`. The page title is "AROGYA". The main content area has a dark grey background and features a large, bold heading "Patient Care Centre". Below the heading is a paragraph of text describing the International Patient Care Centre. To the right of the text is a photograph of a hospital ward with multiple beds and medical equipment. At the bottom of the content area is a "Read More" link. The footer of the page contains copyright information and social media icons.

International Patient Care Centre is a dedicated centre at Lanka Hospitals designed to provide personalized, quality care for international patients travelling to Sri Lanka from overseas. The centre offers a plethora of services ranging from attentive patient care and monitoring to travel and accommodation arrangements, liaising with relevant parties for visa arrangements and much more. We ensure that you are well looked after by our team of trained health professionals.

> Read More

CopyRight © 2022 Apex Design Works

Figure 3. 67 Final Design of Blog.php part 2

```
1 | <?php
2 |
3 | session_start();
4 | if( !isset($_SESSION['User'])) {
5 |     header("location:../../INDEX.php");
6 | }
7 |
8 | ?>
9 |
10| <!-- Header -->
11| <?php include "../../header.php"?>
12|
13| <!-- ----- Nav Bar
14| ----->
15| <nav class="navbar navbar-light bg-info navbar-expand-sm fixed-top">
16|     <div class="container">
17|         <a href="#" class="navbar-brand" > AROGYA </a>
18|
19|         <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavDropdown">
20|             <span class="navbar-toggler-icon"></span> </button>
21|
22|         <div class="collapse navbar-collapse" id="navbarNavDropdown" > <!-- Nav Bar can collapse
23| like a drop down button -->
24|
25|             <ul class="navbar-nav ms-auto">
26|                 <li class="nav-item"> <a href="Home.php" class="nav-link" > Home </a>
27|                 </li>
28|                 <li class="nav-item"> <a href="About.php" class="nav-link" > About Us </a>
29|                 </li>
30|                 <li class="nav-item"> <a href="Blog.php" class="nav-link active" > Blog </a>
31|                 </li>
32|                 <li class="nav-item"> <a href="Contact.php" class="nav-link" > Contact Us </a>
33|             </li>
34|
35|             <li class="nav-item dropdown">
36|                 <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-toggle="dropdown" > Services </a>
37|
38|                 <div class="dropdown-menu">
39|                     <a class="dropdown-item" href="../Staff.php">Staff</a>
40|                     <a class="dropdown-item" href="../Patients.php">Patients</a>
41|                     <a class="dropdown-item" href="../Wards.php">Wards</a>
42|                     <a class="dropdown-item" href="../Rooms.php">Rooms</a>
43|                 </div>
44|             </li>
45|
46|             <li class="nav-item dropdown">
47|                 <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-toggle="dropdown" > Supplies </a>
48|             </li>
49|         </ul>
50|     </div>
51| </nav>
```

Figure 3. 68 Coding for Blog.php part I

```

43             <div class="dropdown-menu">
44                 <a class="dropdown-item" href="#">Medicines</a>
45                 <a class="dropdown-item" href="#">Other</a>
46             </div>
47         </li>
48     </ul>
49   </div>
50 
51 </nav>
52 
53 <br><br><br>
54 
55 <!-- Show Case -> Get Foundation
56 
57 <!-->
58 
59 <!-->
60 <!-->
61 <section class="p-5" id="Learn1">
62     <div class="container">
63         <div class="row align-items-center justify-content-between">
64             <div class="col-md">
65                 
66             </div>
67             <div class="col-md p-5">
68                 <h1 class="text-center">COVID-19 Treatment </h1>
69                 <p class="lead text-center">We offer level 1 care in four luxury hotels in the
70                     vicinity of Lanka Hospitals for asymptomatic & mildly symptomatic patients, with 24/7 hour
71                     medical & nursing care under the supervision of specialised consultants with advanced High
72                     Dependency Care. </p>
73 
74                 <p class="lead text-center">We provide level 2 care in a purpose-built COVID-19 ICU
75                     with critical care beds dedicated for patients who need ICU care with the best critical care
76                     doctors & nurses, ventilator support, 24-hour monitoring, high-flow oxygen, all
77                     investigations and medication.</p>
78 
79                 <button class="btn btn-light mt-3">
80                     <i class="bi bi-chevron-right"></i>
81                     Read More
82                 </button>
83             </div>
84         </div>
85     </div>
86 </section>
87

```

Figure 3. 69 Coding for Blog.php part 2

```

87
88 </section>
89
90 <!-- Show Case -> Get Discounts
91
92 <section class="p-5" id="Discount1">
93
94   <div class="container">
95
96     <div class="row align-items-center justify-content-between bg-dark text-light">
97
98       <div class="col-md p-5">
99         <h1 class="text-center"> International Patient Care Centre </h1>
100        <p class="lead text-center">International Patient Care Centre is a dedicated centre at Lanka Hospitals designed to provide personalized, quality care for international patients travelling to Sri Lanka from overseas. The centre offers a plethora of services ranging from attentive patient care and monitoring to travel and accommodation arrangements, liaising with relevant parties for visa arrangements and much more. We ensure that you are well looked after by our team of trained health professionals. </p>
101
102        <button class="btn btn-dark mt-3">
103          <i class="bi bi-chevron-right"></i>
104          Read More
105        </button>
106      </div>
107
108      <div class="col-md">
109        
110      </div>
111
112    </div>
113  </div>
114
115 </section>
116
117
118 <!-- Footer part
119
120 <footer class="p-5 bg-dark text-white text-center position-relative">
121
122   <div class="container">
123     <p class="lead">CopyRight © 2022 Apex Design Works</p>
124     <p class="lead"> <i class="bi bi-person-bounding-box"></i> </p>
125
126     <a href="#" class="position-absolute bottom-0 end-0 p-5" >
127       <i class="bi bi-arrow-up-circle h1"></i>
128     </a>
129   </div>
130 </footer>

```

Figure 3. 70 Coding for Blog.php part 3

Contact Us

Contact.php :

The screenshot shows a web browser window with the URL localhost/Arogya/Page_Files/Details_Pages/Contact.php. The page has a teal header bar with the logo 'AROGYA' and navigation links: Home, About Us, Blog, Contact Us, Services, and Supplies. Below the header is a section titled 'Frequently Asked Questions'. It contains several expandable questions:

- What procedures will I have in the Emergency Department? (The content is visible: 'Treatments and procedures may include lab work, radiological studies, CAT scans, ultrasound, EKG for cardiac review, medication administration and physician evaluations.')
[Expand/Collapse]
- How is the Emergency Department staffed?
[Expand/Collapse]
- Will I be allowed to have visitors during my treatment?
[Expand/Collapse]
- What happens if I am admitted?
[Expand/Collapse]

Below this section is a heading 'Contact Info' followed by the main location details. To the right is a map showing the location of the South Asian Institute of Technology and Faculty of Computing in Pittugala, Colombo. A copyright notice at the bottom states 'CopyRight © 2022 Apex Design Works'.

Contact Info

Main Location: 578 Elvitigala
Mawatha, Colombo 00500

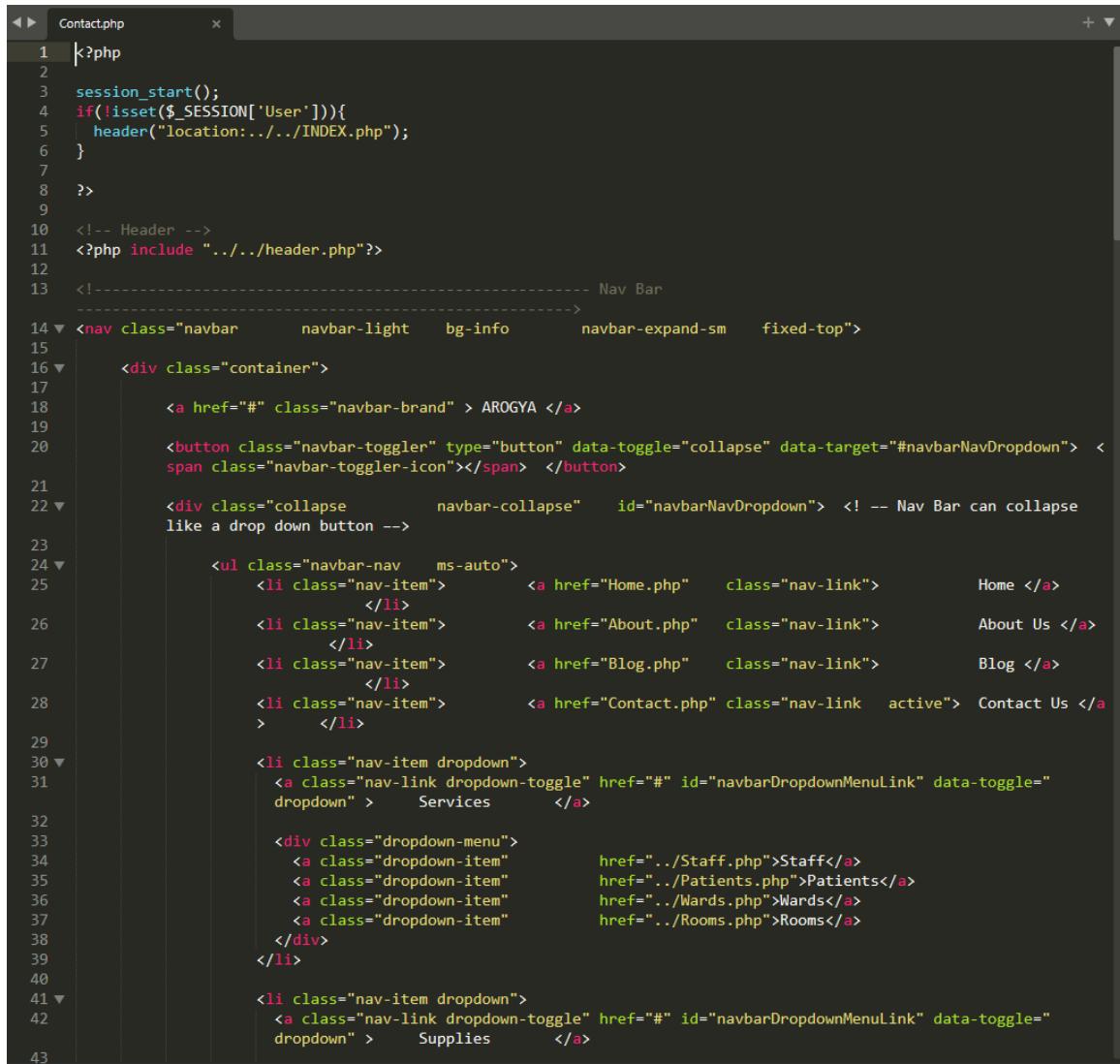
Phone: +94 11 754 4801

Office Email: INFO@AROGYA.LK

Map showing the location of the South Asian Institute of Technology and Faculty of Computing in Pittugala, Colombo. The map includes labels for Lake Road and E002.

CopyRight © 2022 Apex Design Works

Figure 3. 71 Final Design of Contact.php

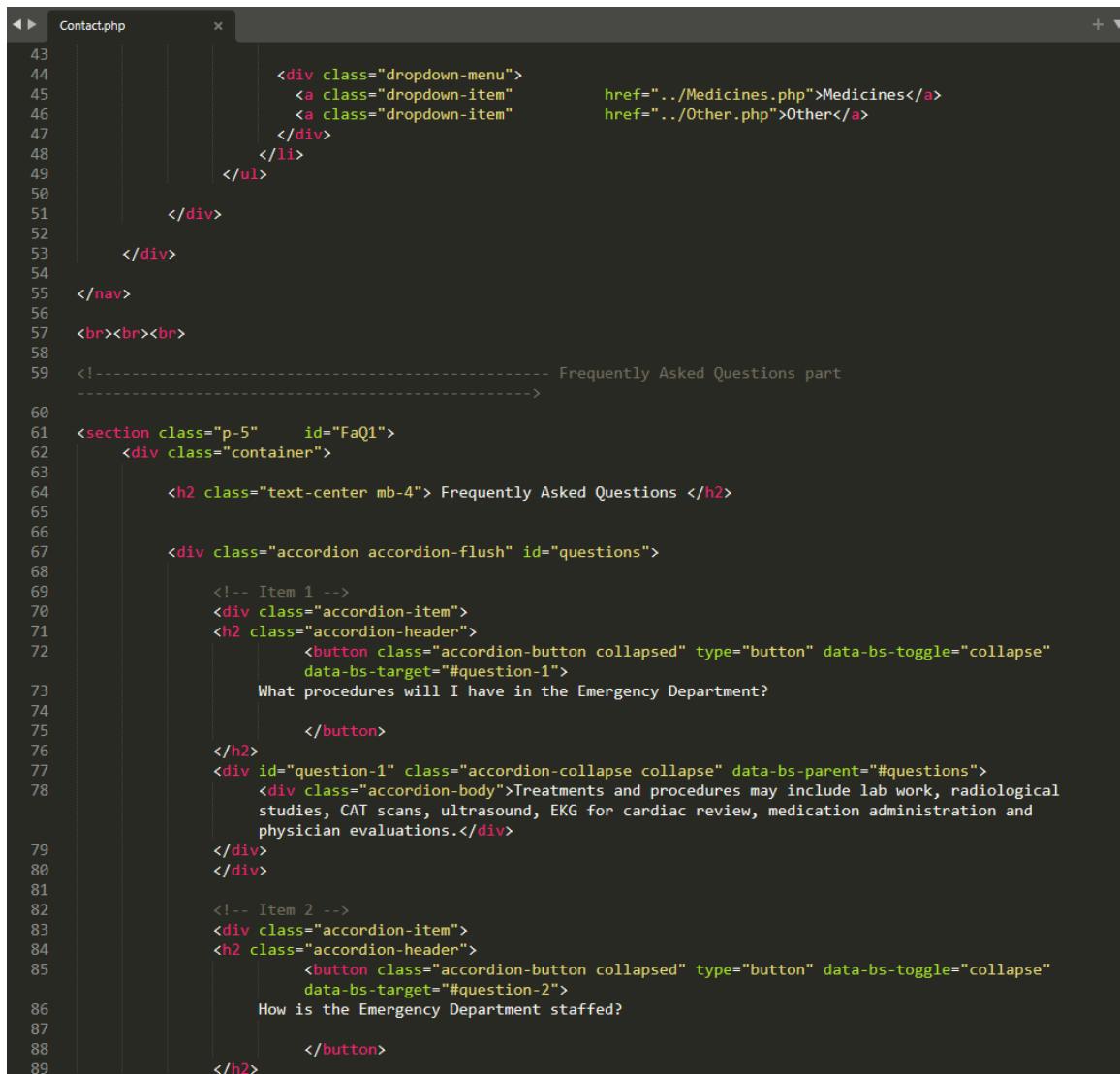


```

1 <?php
2
3 session_start();
4 if(!isset($_SESSION['User'])){
5 header("location:../../INDEX.php");
6 }
7
8 ?>
9
10 <!-- Header -->
11 <?php include "../../header.php"?>
12
13 <!-- ----- Nav Bar
14 <nav class="navbar navbar-light bg-info navbar-expand-sm fixed-top">
15
16 <div class="container">
17
18 <a href="#" class="navbar-brand" > AROGYA </a>
19
20 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavDropdown"> <
21
22 <div class="collapse navbar-collapse" id="navbarNavDropdown" > <!-- Nav Bar can collapse
like a drop down button -->
23
24 <ul class="navbar-nav ms-auto">
25 <li class="nav-item" > <a href="Home.php" class="nav-link" > Home </a>
26 <li class="nav-item" > <a href="About.php" class="nav-link" > About Us </a>
27 <li class="nav-item" > <a href="Blog.php" class="nav-link" > Blog </a>
28 <li class="nav-item" > <a href="Contact.php" class="nav-link active" > Contact Us </a>
29 </li>
30 <li class="nav-item dropdown" >
31 <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-toggle="dropdown" > Services </a>
32
33 <div class="dropdown-menu" >
34 <a class="dropdown-item" href="../Staff.php" > Staff </a>
35 <a class="dropdown-item" href="../Patients.php" > Patients </a>
36 <a class="dropdown-item" href="../Wards.php" > Wards </a>
37 <a class="dropdown-item" href="../Rooms.php" > Rooms </a>
38 </div>
39 </li>
40
41 <li class="nav-item dropdown" >
42 <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-toggle="dropdown" > Supplies </a>
43

```

Figure 3. 72 Coding for Contact.php part 1

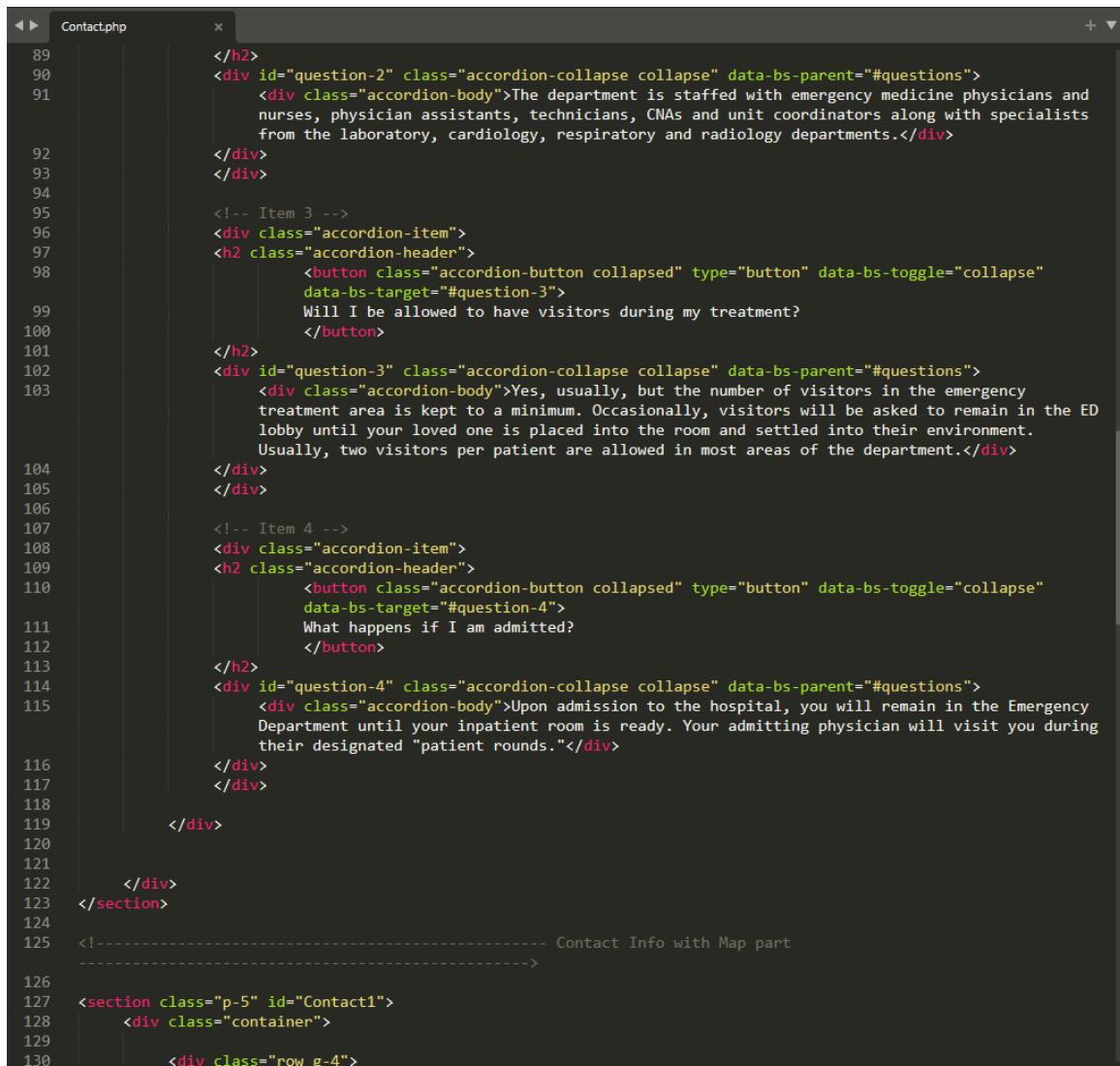


```

43             <div class="dropdown-menu">
44                 <a class="dropdown-item" href="../Medicines.php">Medicines</a>
45                 <a class="dropdown-item" href="../Other.php">Other</a>
46             </div>
47         </li>
48     </ul>
49   </div>
50 
51 </div>
52 
53 <br><br><br>
54 
55 <!-- Frequently Asked Questions part
56 ----->
57 
58 <section class="p-5" id="FaQ1">
59     <div class="container">
60         <h2 class="text-center mb-4"> Frequently Asked Questions </h2>
61 
62         <div class="accordion accordion-flush" id="questions">
63 
64             <!-- Item 1 -->
65             <div class="accordion-item">
66                 <h2 class="accordion-header">
67                     <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse"
68                         data-bs-target="#question-1">
69                         What procedures will I have in the Emergency Department?
70                     </button>
71                 </h2>
72                 <div id="question-1" class="accordion-collapse collapse" data-bs-parent="#questions">
73                     <div class="accordion-body">Treatments and procedures may include lab work, radiological
74                         studies, CAT scans, ultrasound, EKG for cardiac review, medication administration and
75                         physician evaluations.</div>
76                 </div>
77             </div>
78         </div>
79 
80         <!-- Item 2 -->
81         <div class="accordion-item">
82             <h2 class="accordion-header">
83                 <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse"
84                     data-bs-target="#question-2">
85                         How is the Emergency Department staffed?
86                     </button>
87                 </h2>
88             </div>
89         </div>

```

Figure 3.73 Coding for Contact.php part 2



```

89     </h2>
90     <div id="question-2" class="accordion-collapse collapse" data-bs-parent="#questions">
91         <div class="accordion-body">The department is staffed with emergency medicine physicians and
92             nurses, physician assistants, technicians, CNAs and unit coordinators along with specialists
93             from the laboratory, cardiology, respiratory and radiology departments.</div>
94     </div>
95
96     <!-- Item 3 -->
97     <div class="accordion-item">
98         <h2 class="accordion-header">
99             <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse"
100                 data-bs-target="#question-3">
101                 Will I be allowed to have visitors during my treatment?
102             </button>
103         </h2>
104         <div id="question-3" class="accordion-collapse collapse" data-bs-parent="#questions">
105             <div class="accordion-body">Yes, usually, but the number of visitors in the emergency
106                 treatment area is kept to a minimum. Occasionally, visitors will be asked to remain in the ED
107                 lobby until your loved one is placed into the room and settled into their environment.
108                 Usually, two visitors per patient are allowed in most areas of the department.</div>
109         </div>
110     </div>
111
112     <!-- Item 4 -->
113     <div class="accordion-item">
114         <h2 class="accordion-header">
115             <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse"
116                 data-bs-target="#question-4">
117                 What happens if I am admitted?
118             </button>
119         </h2>
120         <div id="question-4" class="accordion-collapse collapse" data-bs-parent="#questions">
121             <div class="accordion-body">Upon admission to the hospital, you will remain in the Emergency
122                 Department until your inpatient room is ready. Your admitting physician will visit you during
123                 their designated "patient rounds."</div>
124         </div>
125     </div>
126
127     <!-- ----- Contact Info with Map part
128     ----->
129     <section class="p-5" id="Contact1">
130         <div class="container">
131             <div class="row g-4">

```

Figure 3. 74 Coding for Contact.php part 3

```
130     <div class="row g-4">
131         <div class="col-md">
132             <h2 class="text-center mb-4">Contact Info</h2>
133             <ul class="list-group list-group-flush lead">
134                 <li class="list-group-item"> <span class="fw-bold">Main Location:</span> 578
135                     Elvitigala Mawatha, Colombo 00500 </li>
136                     <li class="list-group-item"> <span class="fw-bold">Phone:</span> +94 11 754
137                     4801 </li>
138                     <li class="list-group-item"> <span class="fw-bold">Office Email:</span>
139                         INFO@AROGYA.LK </li>
140             </ul>
141         </div>
142         <div class="col-md">
143             <div id="map"></div>
144         </div>
145     </div>
146 </section>
147
148 <!-- JS CDN bundle -->
149 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYs0g+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p" crossorigin="anonymous"></script>
150
151 <!-- JS Mapbox -->
152 <script src='https://api.mapbox.com/mapbox-gl-js/v2.8.0/mapbox-gl.js'></script>
153
154 <script>
155    .mapboxgl.accessToken = 'pk.eyJ1IjoicnlhbndpY2tyYW1hcmF0bmUiLCJhIjoiY2wxc3o2aWh1MDM3dTNkbnkybjc2d2ppcyJ9.xiN3YL3nFTM63Ib9X1B4Ig';
156     var map = new mapboxgl.Map({
157         container: 'map',
158         style: 'mapbox://styles/mapbox/streets-v11',
159         center: [79.9729, 6.9147],
160         zoom: 13
161     });
162
163 })();
164
165 </script>
166
167
168 <!------- Footer part
----->
169 <footer class="p-5 bg-dark    text-white    text-center    position-relative">
170     <div class="container">
171         <p class="lead">CopyRight © 2022 Apex Design Works</p>
172         <p class="lead"> <i class="bi bi-person-bounding-box"></i> </p>
173     </div>
174 
```

Figure 3. 75 Coding for Contact.php part 4



```
167 <!------- Footer part ----->
168 
169 <footer class="p-5 bg-dark    text-white  text-center      position-relative">
170     <div class="container">
171         <p class="lead">CopyRight © 2022 Apex Design Works</p>
172         <p class="lead"> <i class="bi bi-person-bounding-box"></i> </p>
173 
174         <a href="" class="position-absolute    bottom-0  end-0    p-5"> <i class="bi bi-arrow-up-circle h1"></i> </a>
175     </div>
176 
177 </footer>
178 
179 
180 
181 
182 
183 
184 
185 
```

Figure 3. 76 Coding for Contact.php part 5

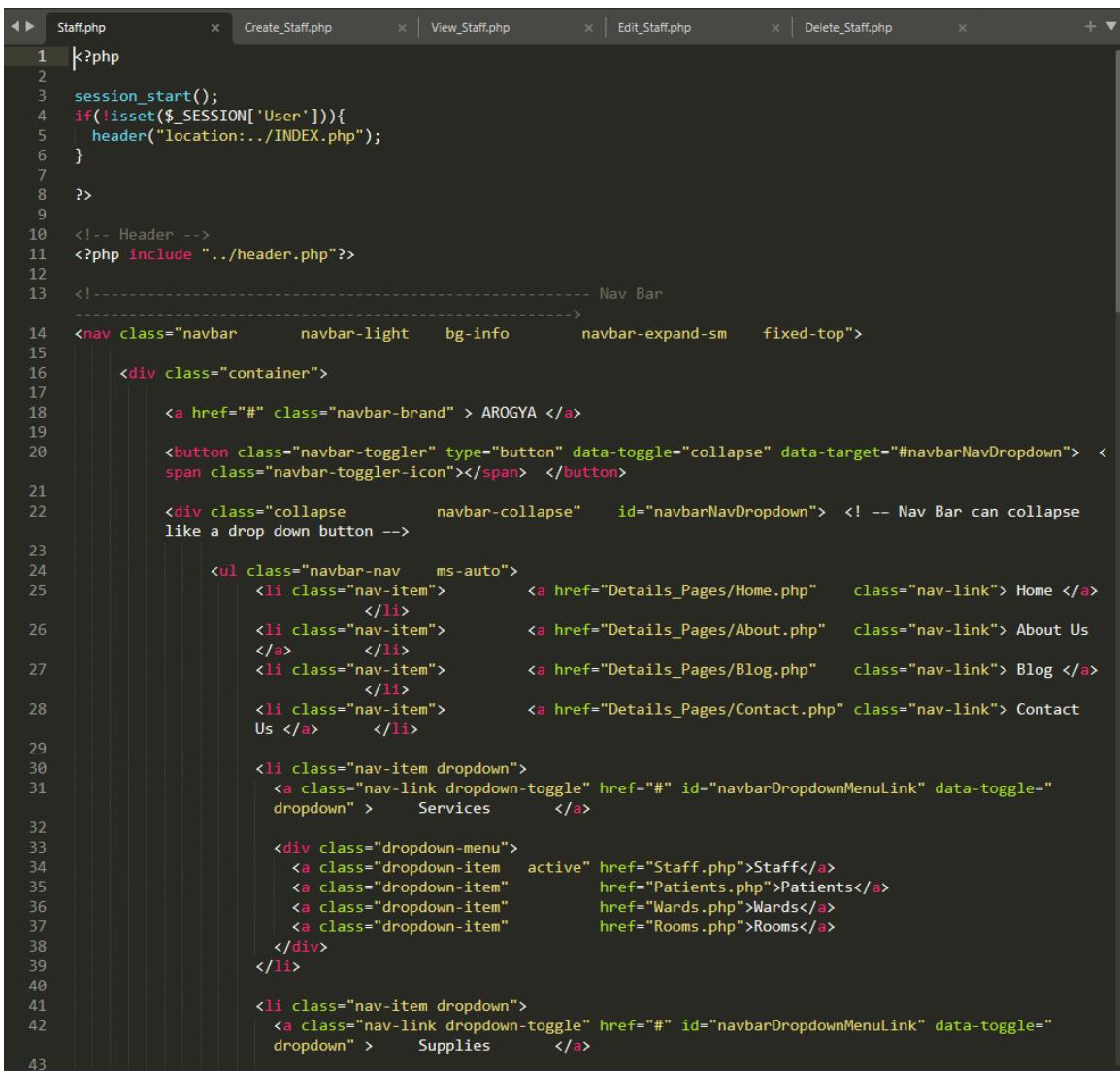
Staff Page

Staff.php :

| ID Number | Name | Type | Options | | |
|-----------|--------------------|------------|---------|------|--------|
| 1 | Pavel Fernandoooo | Doctor | View | Edit | Delete |
| 2 | Ryan Wickramaratne | Nurse | View | Edit | Delete |
| 3 | Judith Michelle | Technician | View | Edit | Delete |
| 4 | Kyle Wickramaratne | Attendant | View | Edit | Delete |
| 5 | Sadew Fernando | Doctor | View | Edit | Delete |
| 6 | Mashi Perera | Nurse | View | Edit | Delete |
| 7 | Shyni Perera | Attendant | View | Edit | Delete |

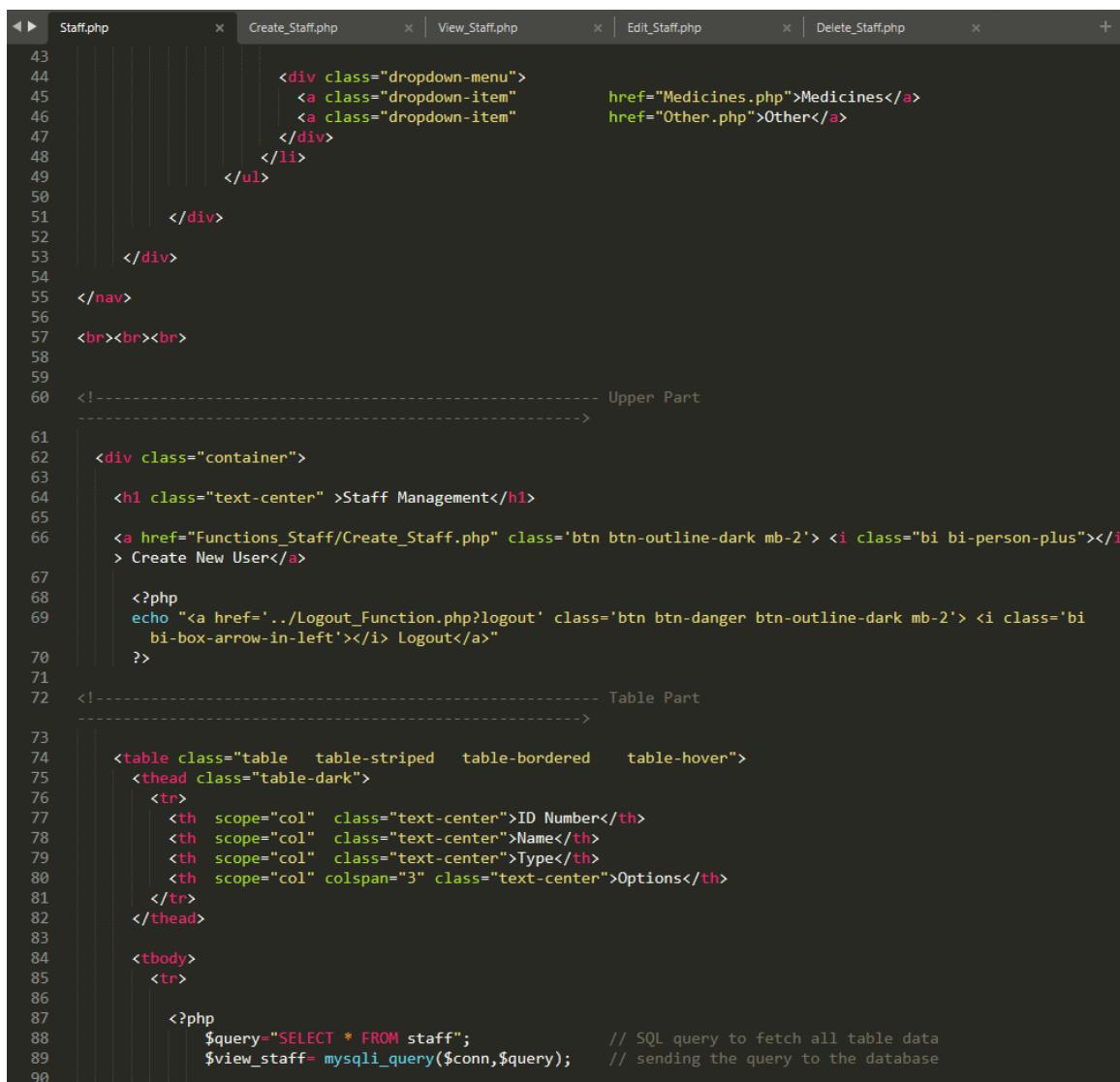
[Back](#)

Figure 3. 77 Final Design of Staff.php



```
1 <?php
2
3 session_start();
4 if(!isset($_SESSION['User'])){
5 header("location:../INDEX.php");
6 }
7
8 ?>
9
10 <!-- Header -->
11 <?php include "../header.php"?>
12
13 <!-- Nav Bar
14 <nav class="navbar navbar-light bg-info navbar-expand-sm fixed-top">
15
16     <div class="container">
17
18         <a href="#" class="navbar-brand" > AROGYA </a>
19
20         <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavDropdown"> <span class="navbar-toggler-icon"></span> </button>
21
22         <div class="collapse navbar-collapse" id="navbarNavDropdown" > ! -- Nav Bar can collapse
23             like a drop down button -->
24
25             <ul class="navbar-nav ms-auto">
26                 <li class="nav-item" > <a href="Details_Pages/Home.php" class="nav-link" > Home </a>
27                     </li>
28                 <li class="nav-item" > <a href="Details_Pages/About.php" class="nav-link" > About Us </a>
29                     </li>
30                 <li class="nav-item" > <a href="Details_Pages/Blog.php" class="nav-link" > Blog </a>
31                     </li>
32                 <li class="nav-item" > <a href="Details_Pages/Contact.php" class="nav-link" > Contact Us </a>
33                     </li>
34
35             <li class="nav-item dropdown">
36                 <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-toggle="dropdown" > Services </a>
37
38                 <div class="dropdown-menu">
39                     <a class="dropdown-item active" href="Staff.php" >Staff</a>
40                     <a class="dropdown-item" href="Patients.php" >Patients</a>
41                     <a class="dropdown-item" href="Wards.php" >Wards</a>
42                     <a class="dropdown-item" href="Rooms.php" >Rooms</a>
43                 </div>
44             </li>
45
46             <li class="nav-item dropdown">
47                 <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-toggle="dropdown" > Supplies </a>
48             </li>
49         </ul>
50     </div>
51 </nav>
```

Figure 3. 78 Coding for Staff.php part 1

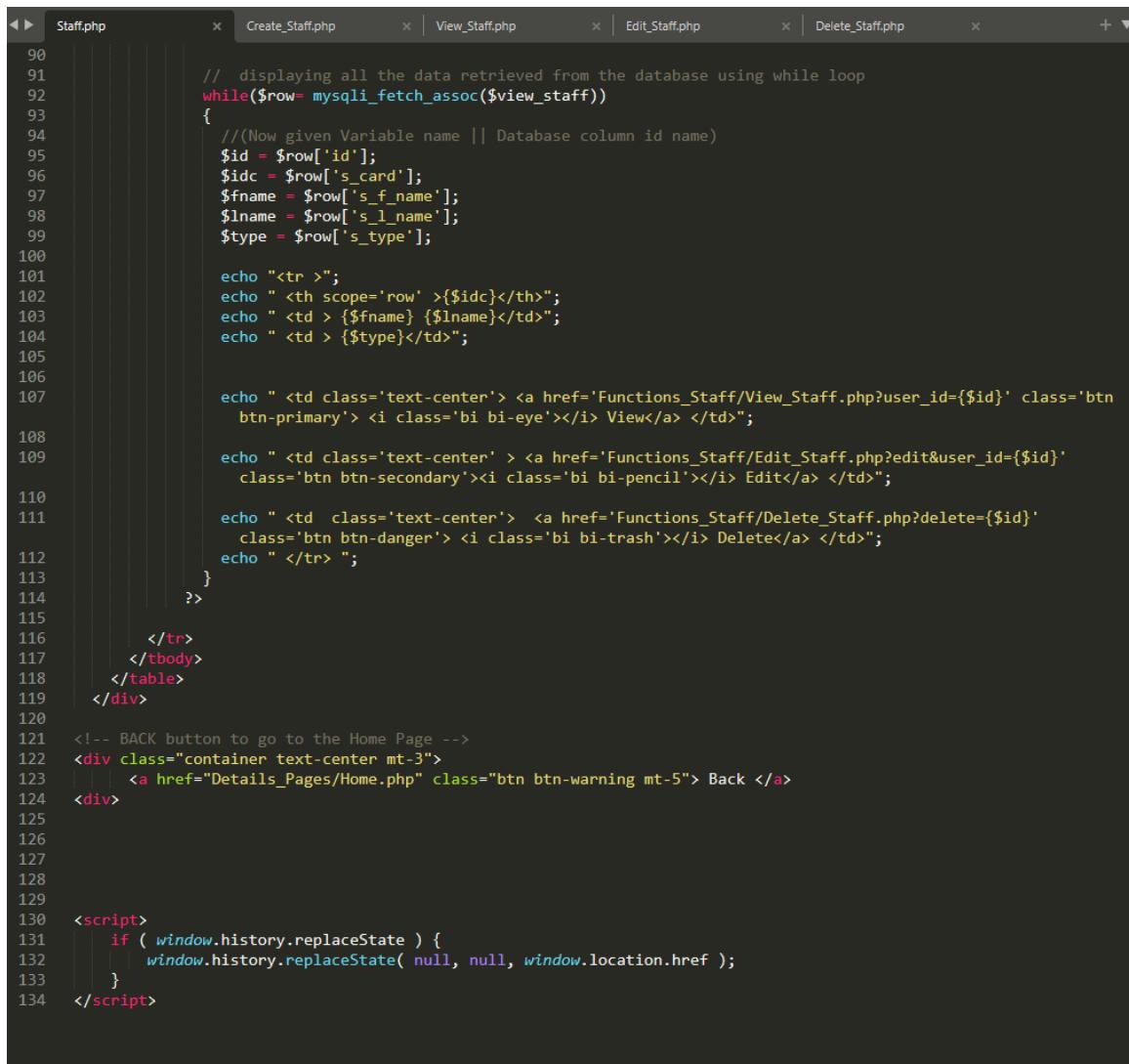


```

43             <div class="dropdown-menu">
44                 <a class="dropdown-item" href="Medicines.php">Medicines</a>
45                 <a class="dropdown-item" href="Other.php">Other</a>
46             </div>
47         </li>
48     </ul>
49 </div>
50
51     <br><br><br>
52
53 </div>
54
55 </nav>
56
57 <br><br><br>
58
59
60 <!-- Upper Part -->
61
62 <div class="container">
63
64     <h1 class="text-center" >Staff Management</h1>
65
66     <a href="Functions_Staff/Create_Staff.php" class='btn btn-outline-dark mb-2'><i class="bi bi-person-plus"></i> Create New User</a>
67
68     <?php
69     echo "<a href='../Logout_Function.php?logout' class='btn btn-danger btn-outline-dark mb-2'><i class='bi bi-box-arrow-in-left'></i> Logout</a>">
70     ?>
71
72 <!-- Table Part -->
73
74 <table class="table table-striped table-bordered table-hover">
75     <thead class="table-dark">
76         <tr>
77             <th scope="col" class="text-center">ID Number</th>
78             <th scope="col" class="text-center">Name</th>
79             <th scope="col" class="text-center">Type</th>
80             <th scope="col" colspan="3" class="text-center">Options</th>
81         </tr>
82     </thead>
83
84     <tbody>
85         <tr>
86
87             <?php
88                 $query="SELECT * FROM staff";           // SQL query to fetch all table data
89                 $view_staff= mysqli_query($conn,$query); // sending the query to the database
90             </tr>

```

Figure 3. 79 Coding for Staff.php part 1



```

90
91     // displaying all the data retrieved from the database using while loop
92     while($row= mysqli_fetch_assoc($view_staff))
93     {
94         //Now given Variable name || Database column id name
95         $id = $row['id'];
96         $idc = $row['s_card'];
97         $fname = $row['s_f_name'];
98         $lname = $row['s_l_name'];
99         $type = $row['s_type'];
100
101        echo "<tr >";
102        echo " <th scope='row' >{$idc}</th>";
103        echo " <td > {$fname} {$lname}</td>";
104        echo " <td > {$type}</td>";
105
106
107        echo " <td class='text-center'> <a href='Functions_Staff/View_Staff.php?user_id={$id}' class='btn
108             btn-primary'> <i class='bi bi-eye'></i> View</a> </td> ";
109        echo " <td class='text-center'> <a href='Functions_Staff/Edit_Staff.php?edit&user_id={$id}'>
110             <i class='bi bi-pencil'></i> Edit</a> </td> ";
111
112        echo " <td class='text-center'> <a href='Functions_Staff/Delete_Staff.php?delete={$id}'>
113             <i class='bi bi-trash'></i> Delete</a> </td> ";
114    }
115
116    </tr>
117  </tbody>
118 </table>
119 </div>
120
121 <!-- BACK button to go to the Home Page -->
122 <div class="container text-center mt-3">
123   <a href="Details_Pages/Home.php" class="btn btn-warning mt-5"> Back </a>
124 </div>
125
126
127
128
129
130 <script>
131   if ( window.history.replaceState ) {
132     window.history.replaceState( null, null, window.location.href );
133   }
134 </script>

```

Figure 3. 80 Coding for Staff.php part 2

Create_Staff.php :

The screenshot shows a web browser window with the URL `localhost/Arogya/Page_Files/Functions_Staff/Create_Staff.php`. The page title is "Add Staff Details". The form consists of several input fields arranged in a grid:

| ID Number | First Name |
|-----------------|--|
| Enter ID Number | Enter First Name |
| Last Name | Mobile |
| Enter Last Name | Enter Mobile Number |
| Address | Gender |
| Enter Address | Choose... |
| Type | Date of Birth |
| Choose... | mm/dd/yyyy <input type="button" value=""/> |

Below the form are two buttons: a blue "Submit" button and a yellow "Back" button.

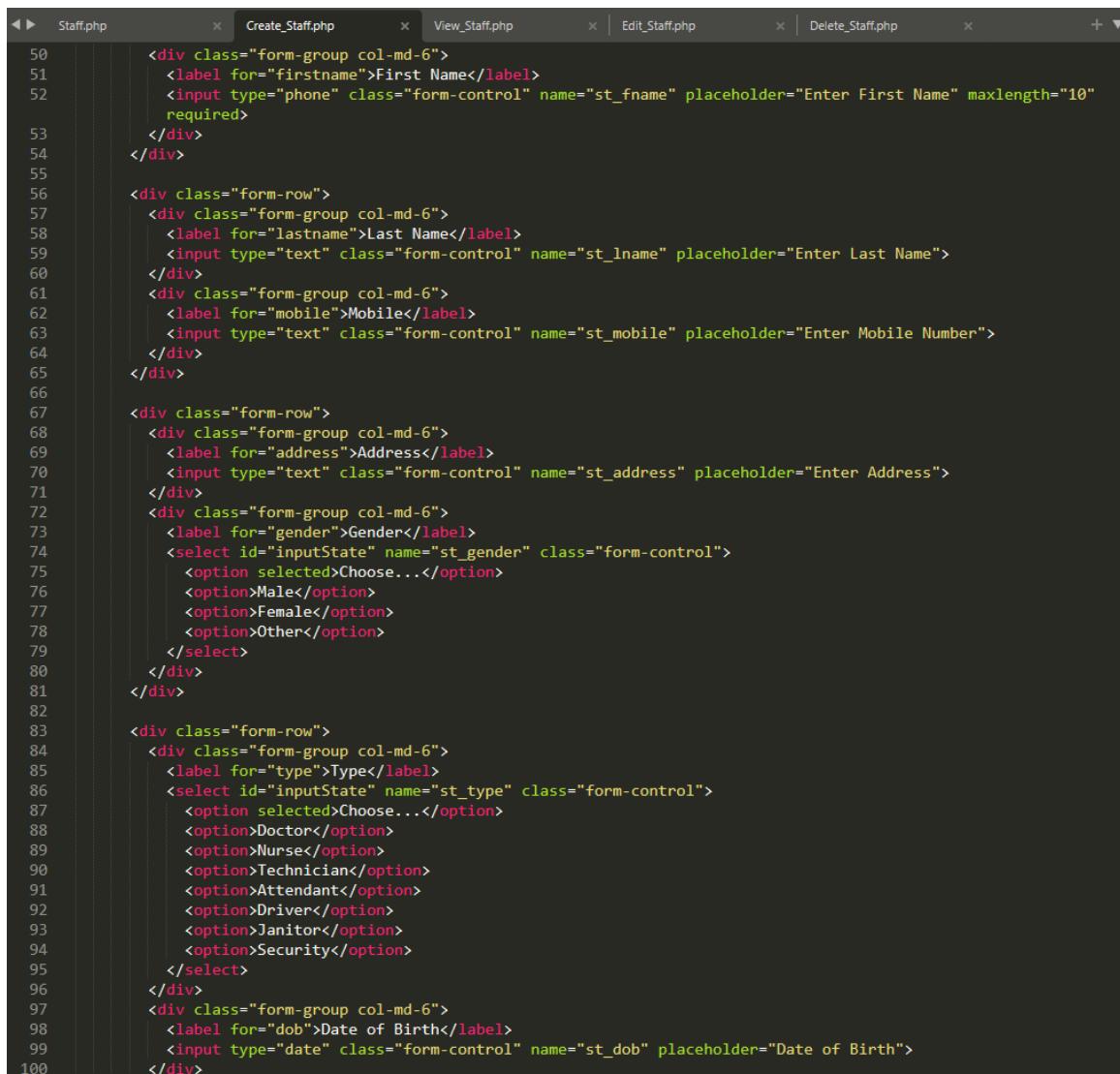
Figure 3. 81 Final Design of Create_Staff.php

```

1 | <!-- Header -->
2 | <?php include "../header.php" ?>
3 |
4 | <?php
5 |
6 | //To give new values to DATABASE
7 | if(isset($_POST['submit']))
8 | {
9 |     //Now given Variable name || Texbox id name
10 |     $id = $_POST['card_no'];
11 |     $sfname = $_POST['st_fname'];
12 |     $slname = $_POST['st_lname'];
13 |     $sphone = $_POST['st_mobile'];
14 |     $saddress = $_POST['st_address'];
15 |     $sgender = $_POST['st_gender'];
16 |     $stype = $_POST['st_type'];
17 |     $sdob = $_POST['st_dob'];
18 |
19 |
20 |
21 |     // SQL query to insert user data into the users table
22 |     //Database column id name || Previously given Variable name)
23 |     $query= "INSERT INTO staff(s_card, s_f_name, s_l_name, s_phone, s_address, s_gender, s_type, s_dob) VALUES('{$id}', '{$sfname}', '{$slname}', '{$sphone}', '{$saddress}', '{$sgender}', '{$stype}', '{$sdob}')";
24 |     $add_user = mysqli_query($conn,$query);
25 |
26 |     // displaying proper message for the user to see whether the query executed perfectly or not
27 |     if (!$add_user)
28 |     {
29 |         echo "something went wrong ". mysqli_error($conn);
30 |     }
31 |
32 |     else
33 |     { echo "<script type='text/javascript'>alert('Staff member added successfully!')</script>";
34 |       header('Location:../Staff.php');
35 |     }
36 |
37 | }
38 ?>
39 |
40 |
41 | <h1 class="text-center">Add Staff Details </h1>
42 | <div class="container">
43 |     <form action="" method="post">
44 |
45 |         <div class="form-row">
46 |             <div class="form-group col-md-6">
47 |                 <label for="idnumber">ID Number</label>
48 |                 <input type="text" class="form-control" name="card_no" placeholder="Enter ID Number" maxlength="12"
49 |                     required>
50 |             </div>
51 |             <div class="form-group col-md-6">

```

Figure 3. 82 Coding for Create_Staff.php part 1



```

50     <div class="form-group col-md-6">
51         <label for="firstname">First Name</label>
52         <input type="phone" class="form-control" name="st_fname" placeholder="Enter First Name" maxlength="10"
53             required>
54     </div>
55
56     <div class="form-row">
57         <div class="form-group col-md-6">
58             <label for="lastname">Last Name</label>
59             <input type="text" class="form-control" name="st_lname" placeholder="Enter Last Name">
60         </div>
61         <div class="form-group col-md-6">
62             <label for="mobile">Mobile</label>
63             <input type="text" class="form-control" name="st_mobile" placeholder="Enter Mobile Number">
64         </div>
65     </div>
66
67     <div class="form-row">
68         <div class="form-group col-md-6">
69             <label for="address">Address</label>
70             <input type="text" class="form-control" name="st_address" placeholder="Enter Address">
71         </div>
72         <div class="form-group col-md-6">
73             <label for="gender">Gender</label>
74             <select id="inputState" name="st_gender" class="form-control">
75                 <option selected>Choose...</option>
76                 <option>Male</option>
77                 <option>Female</option>
78                 <option>Other</option>
79             </select>
80         </div>
81     </div>
82
83     <div class="form-row">
84         <div class="form-group col-md-6">
85             <label for="type">Type</label>
86             <select id="inputState" name="st_type" class="form-control">
87                 <option selected>Choose...</option>
88                 <option>Doctor</option>
89                 <option>Nurse</option>
90                 <option>Technician</option>
91                 <option>Attendant</option>
92                 <option>Driver</option>
93                 <option>Janitor</option>
94                 <option>Security</option>
95             </select>
96         </div>
97         <div class="form-group col-md-6">
98             <label for="dob">Date of Birth</label>
99             <input type="date" class="form-control" name="st_dob" placeholder="Date of Birth">
100        </div>

```

Figure 3. 83 Coding for Create_Staff.php part 2

```
100 |     </div>
101 |   </div>
102 |
103 |   <div class="col text-center">
104 |     <input type="submit" name="submit" class="btn btn-info btn-large" value="Submit">
105 |   </div>
106 |
107 |   </form>
108 | </div>
109 |
110 |   <!-- BACK button to go to the Staff page --&gt;
111 |   &lt;div class="container text-center mt-3"&gt;
112 |     &lt;a href="../Staff.php" class="btn btn-warning mt-5"&gt; Back &lt;/a&gt;
113 |   &lt;/div&gt;
114 |
115 |
116 |
117 |
118 |</pre>
```

Figure 3. 84 Coding for Create_Staff.php part 3

View_Staff.php :

| ID | Name | Mobile | Address | Gender | Type | Date of Birth |
|----|----------------------|-------------|-------------|--------|--------|---------------|
| 1 | Pavel Fernandoooo | 31222721111 | Basiyawatta | Male | Doctor | 2022-05-14 |

[Back](#)

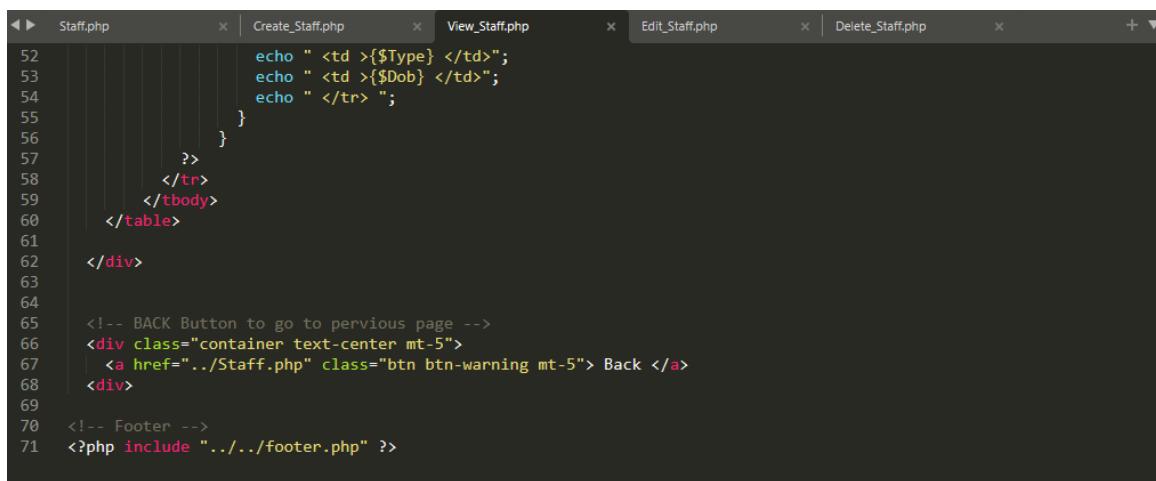
Figure 3. 85 Final Design of View_Staff.php

```

1 <!-- Header -->
2 <?php include '../header.php'?>
3
4 <h1 class="text-center">Staff Details</h1>
5
6 <div class="container">
7
8     <table class="table table-striped table-bordered table-hover">
9         <thead class="table-dark">
10            <tr>
11                <th scope="col">ID</th>
12                <th scope="col">Name</th>
13                <th scope="col">Mobile</th>
14                <th scope="col">Address</th>
15                <th scope="col">Gender</th>
16                <th scope="col">Type</th>
17                <th scope="col">Date of Birth</th>
18            </tr>
19        </thead>
20        <tbody>
21            <tr>
22                <?php
23                    // first we check using 'isset()' function if the variable is set or not
24                    //Processing form data when form is submitted
25                    if (isset($_GET['user_id']))
26                {
27                    $userid = $_GET['user_id'];
28
29                    // SQL query to fetch the data where id=$userid & storing data in view_staff
30                    $query="SELECT * FROM staff WHERE id = {$userid} ";
31                    $view_staff= mysqli_query($conn,$query);
32
33                    while($row = mysqli_fetch_assoc($view_staff))
34                {
35                        //Now given Variable name || Database column id name
36                        $id = $row['id'];
37                        $idc = $row['s_card'];
38                        $fName = $row['s_f_name'];
39                        $lName = $row['s_l_name'];
40                        $Mobile = $row['s_phone'];
41                        $Address = $row['s_address'];
42                        $Gender = $row['s_gender'];
43                        $Type = $row['s_type'];
44                        $Dob = $row['s_dob'];
45
46                        echo "<tr >";
47                        echo " <td >{$idc}</td>";
48                        echo " <td > {$fName} {$lName} </td>";
49                        echo " <td > {$Mobile}</td>";
50                        echo " <td >{$Address}</td>";
51                        echo " <td >{$Gender}</td>";
52                        echo " <td >{$Type}</td>":

```

Figure 3. 86 Coding for View_Staff.php part 1



```
52 echo " <td>{$Type} </td>";
53 echo " <td>{$Dob} </td>";
54 echo " </tr> ";
55 }
56     ?>
57     </tr>
58   </tbody>
59 </table>
60
61 </div>
62
63
64
65    <!-- BACK Button to go to previous page -->
66 <div class="container text-center mt-5">
67   <a href="../Staff.php" class="btn btn-warning mt-5"> Back </a>
68 </div>
69
70
71 <?php include "../../footer.php" ?>
```

Figure 3. 87 Coding for View_Staff.php part 2

Edit_Staff.php :

The screenshot shows a web browser window with the URL `localhost/Arogya/Page_Files/Functions_Staff/Edit_Staff.php?edit...`. The page title is "Edit Staff Details". The form contains the following fields:

| | |
|-------------|---------------|
| ID Number | First Name |
| 1 | Pavel |
| Last Name | Mobile |
| Fernandoooo | 31222721111 |
| Address | Gender |
| Basiyawatta | Male |
| Type | Date of Birth |
| Doctor | 05/14/2022 |

Below the form are two buttons: a teal "Edit" button and a yellow "Back" button.

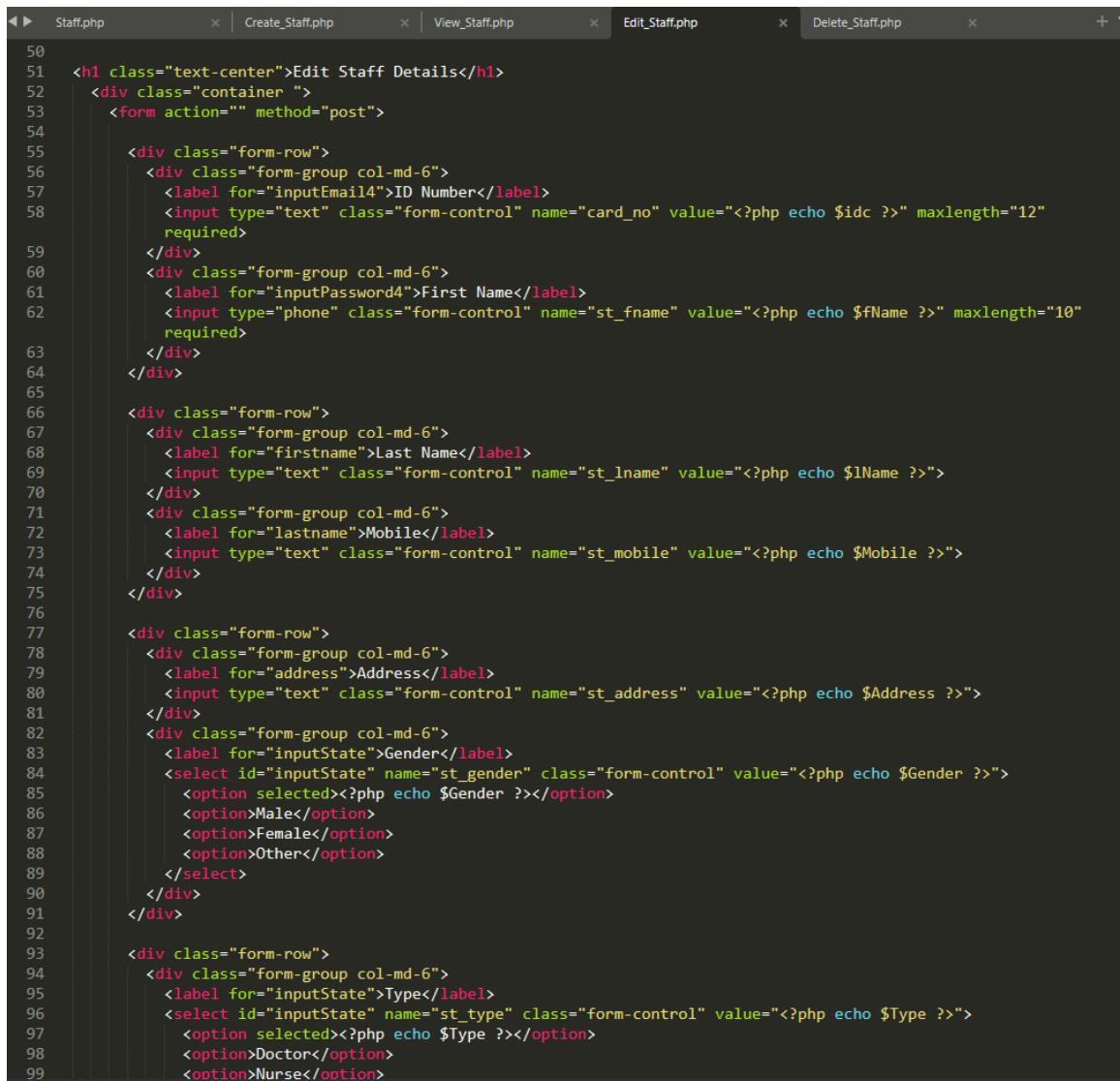
Figure 3. 88 Final Design of Edit_Staff.php

```

1 <!-- Footer -->
2 <?php include "../header.php"?>
3
4 <?php
5 // checking if the variable is set or not and if set adding the set data value to variable userid
6 if(isset($_GET['user_id']))
7 {
8     $userid = $_GET['user_id'];
9 }
10 // SQL query to select all the data from the table where id = $userid ()
11 $query="SELECT * FROM staff WHERE id = $userid ";
12 $edit_staff= mysqli_query($conn,$query);
13
14 //To see DATABASE values
15 while($row = mysqli_fetch_assoc($edit_staff))
16 {
17     //Textbox echo id name || Database column id name
18     $id = $row['id'];
19     $idc = $row['s_card'];
20     $fName = $row['s_f_name'];
21     $lName = $row['s_l_name'];
22     $Mobile = $row['s_phone'];
23     $Address = $row['s_address'];
24     $Gender = $row['s_gender'];
25     $Type = $row['s_type'];
26     $Dob = $row['s_dob'];
27 }
28
29 //To give new values to DATABASE
30 if(isset($_POST['update']))
31 {
32     //Now given Variable name || Textbox id name
33     $sidc = $_POST['card_no'];
34     $sfName = $_POST['st_fname'];
35     $slName = $_POST['st_lname'];
36     $sMobile = $_POST['st_mobile'];
37     $sAddress = $_POST['st_address'];
38     $sGender = $_POST['st_gender'];
39     $sType = $_POST['st_type'];
40     $sDob = $_POST['st_dob'];
41
42     // SQL query to update the data in user table where the id = $userid
43     //(Database column id name || Previously given Variable name)
44     $query = "UPDATE staff SET s_card = '{$sidc}' , s_f_name = '{$sfName}' , s_l_name = '{$slName}' , s_phone =
45         '{$sMobile}' , s_address = '{$sAddress}' , s_gender = '{$sGender}' , s_type = '{$sType}' , s_dob = '{$sDob}'
46         WHERE id = $userid";
47     $update_user = mysqli_query($conn, $query);
48     echo "<script type='text/javascript'>alert('Staff member edited successfully!')</script>";
49     header("location:../Staff.php");
50 }
51 ?>

```

Figure 3. 89 Coding for Edit_Staff.php part 1



```

50 <h1 class="text-center">Edit Staff Details</h1>
51 <div class="container">
52   <form action="" method="post">
53
54     <div class="form-row">
55       <div class="form-group col-md-6">
56         <label for="inputEmail4">ID Number</label>
57         <input type="text" class="form-control" name="card_no" value=<?php echo $idc ?>" maxlength="12"
58           required>
59       </div>
60       <div class="form-group col-md-6">
61         <label for="inputPassword4">First Name</label>
62         <input type="phone" class="form-control" name="st_fname" value=<?php echo $fName ?>" maxlength="10"
63           required>
64       </div>
65     </div>
66
67     <div class="form-row">
68       <div class="form-group col-md-6">
69         <label for="firstname">Last Name</label>
70         <input type="text" class="form-control" name="st_lname" value=<?php echo $lName ?>">
71       </div>
72       <div class="form-group col-md-6">
73         <label for="lastname">Mobile</label>
74         <input type="text" class="form-control" name="st_mobile" value=<?php echo $Mobile ?>">
75       </div>
76     </div>
77
78     <div class="form-row">
79       <div class="form-group col-md-6">
80         <label for="address">Address</label>
81         <input type="text" class="form-control" name="st_address" value=<?php echo $Address ?>">
82       </div>
83       <div class="form-group col-md-6">
84         <label for="inputState">Gender</label>
85         <select id="inputState" name="st_gender" class="form-control" value=<?php echo $Gender ?>">
86           <option selected><?php echo $Gender ?></option>
87           <option>Male</option>
88           <option>Female</option>
89           <option>Other</option>
90         </select>
91       </div>
92
93     <div class="form-row">
94       <div class="form-group col-md-6">
95         <label for="inputState">Type</label>
96         <select id="inputState" name="st_type" class="form-control" value=<?php echo $Type ?>">
97           <option selected><?php echo $Type ?></option>
98           <option>Doctor</option>
99           <option>Nurse</option>

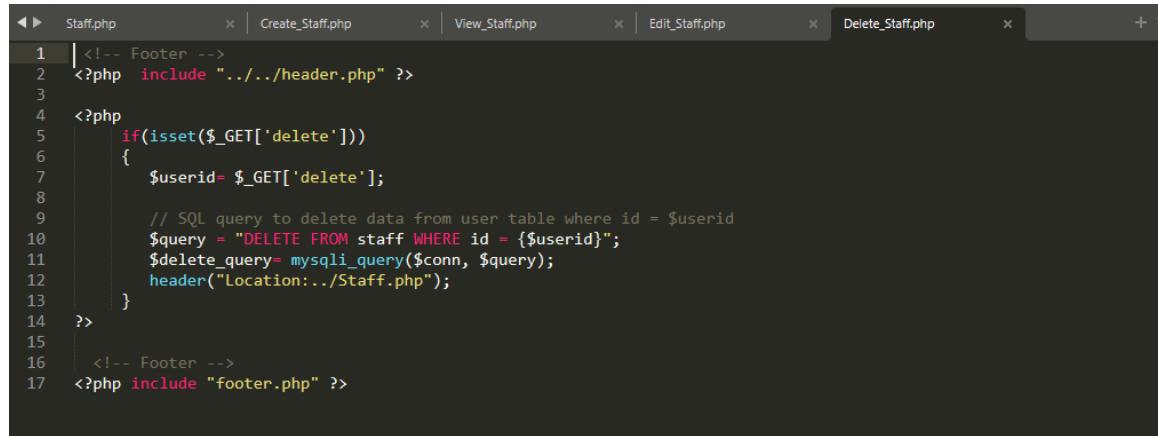
```

Figure 3. 90 Coding for Edit_Staff.php part 2

```
96     <select id="inputState" name="st_type" class="form-control" value=<?php echo $Type ?>>
97         <option selected><?php echo $Type ?></option>
98         <option>Doctor</option>
99         <option>Nurse</option>
100        <option>Technician</option>
101        <option>Attendant</option>
102        <option>Driver</option>
103        <option>Janitor</option>
104        <option>Security</option>
105    </select>
106 </div>
107 <div class="form-group col-md-6">
108     <label for="inputPassword4">Date of Birth</label>
109     <input type="date" class="form-control" name="st_dob" value=<?php echo $Dob ?>>
110 </div>
111 </div>
112 
113     <div class="col text-center">
114         <input type="submit" name="update" class="btn btn-info btn-large" value="Edit">
115     </div>
116 
117 </form>
118 </div>
119 
120     <!-- BACK button to go to the Staff page -->
121 <div class="container text-center mt-5">
122     <a href="../Staff.php" class="btn btn-warning mt-5"> Back </a>
123 </div>
124 
125 
126 
127     <!-- Footer -->
128     <?php include "../footer.php" ?>
```

Figure 3. 91 Coding for Edit_Staff.php part 3

Delete_Staff.php :



```
1 | <!-- Footer -->
2 | <?php include "../header.php" ?>
3 |
4 | <?php
5 |   if(isset($_GET['delete']))
6 |   {
7 |     $userid= $_GET['delete'];
8 |
9 |     // SQL query to delete data from user table where id = $userid
10 |    $query = "DELETE FROM staff WHERE id = {$userid}";
11 |    $delete_query= mysqli_query($conn, $query);
12 |    header("Location:../Staff.php");
13 |  }
14 |
15 |
16 | <!-- Footer -->
17 | <?php include "footer.php" ?>
```

Figure 3. 92 Coding for Delete_Staff.php part 1

Patients Page

Patients.php :

The screenshot shows a web browser window with the URL `localhost/Arogya/Page_Files/Patients.php`. The page title is "AROGYA". The main content area is titled "Patients Management". At the top right of this area are two buttons: "Add New Patient" (with a plus icon) and "Logout" (with a user icon). Below this is a table with the following data:

| ID | Name | Age | Ward | Options | | |
|----|--------------------|-----|--------|----------------------|----------------------|------------------------|
| 1 | Judith Michelle | 20 | Ward 2 | View | Edit | Delete |
| 2 | Ryan Wickramaratne | 25 | Ward 1 | View | Edit | Delete |
| 3 | Pavel Fernando | 15 | Ward 3 | View | Edit | Delete |
| 4 | Malshan Fernando | 25 | Ward 2 | View | Edit | Delete |

Below the table is a green button labeled "Patient Invoice" with a document icon. At the bottom center is a yellow "Back" button.

Figure 3. 93 Final Design of Patients.php

Patient Invoice Modal :

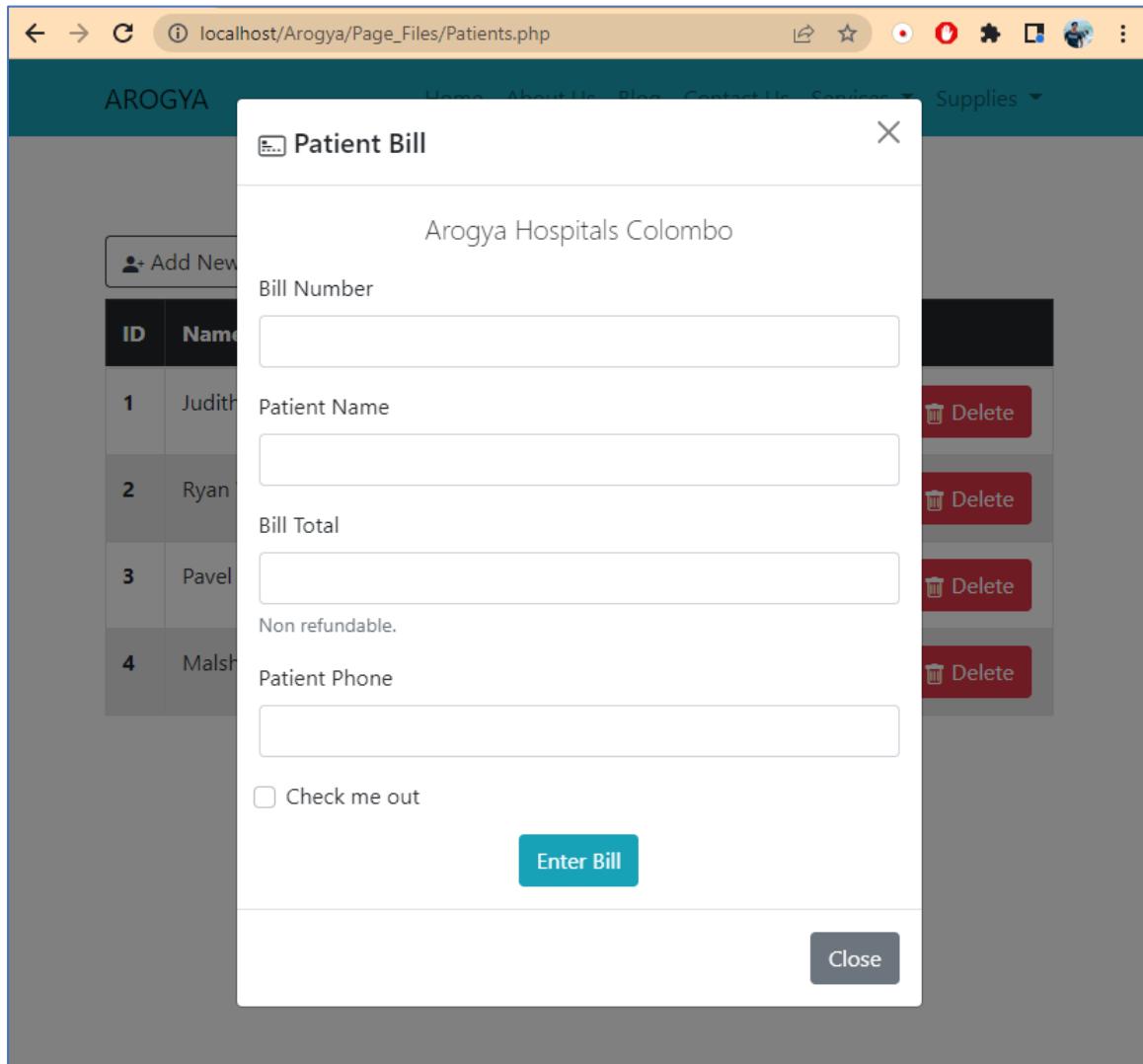


Figure 3. 94 Final Design of Patient Invoice Modal

```
Patients.php x Create_Patient.php x View_Patient.php x Edit_Patient.php x Delete_Patient.php x +
```

```
1 <?php
2
3 session_start();
4 if(!isset($_SESSION['User'])){
5 header("location: index.php");
6 }
7
8 ?>
9
10 <!-- Header -->
11 <?php include "../header.php"?>
12
13
14
15
16
17
18 <!------- Nav Bar----->
19 <nav class="navbar navbar-light bg-info navbar-expand-sm fixed-top">
20
21 <div class="container">
22
23 <a href="#" class="navbar-brand" > AROGYA </a>
24
25 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavDropdown"><span class="navbar-toggler-icon"></span> </button>
26
27 <div class="collapse navbar-collapse" id="navbarNavDropdown" > <!-- Nav Bar can collapse like a drop down button -->
28
29 <ul class="navbar-nav ms-auto">
30 <li class="nav-item"> <a href="Details_Pages/Home.php" class="nav-link"> Home </a>
31 </li>
32 <li class="nav-item"> <a href="Details_Pages/About.php" class="nav-link"> About Us </a>
33 </li>
34 <li class="nav-item"> <a href="Details_Pages/Blog.php" class="nav-link"> Blog </a>
35 </li>
36 <li class="nav-item"> <a href="Details_Pages/Contact.php" class="nav-link"> Contact Us </a> </li>
37
38 <li class="nav-item dropdown">
39 <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-toggle="dropdown" > Services </a>
40
41 <div class="dropdown-menu">
42 <a class="dropdown-item" href="Staff.php">Staff</a>
43 <a class="dropdown-item active" href="Patients.php">Patients</a>
44 <a class="dropdown-item" href="Wards.php">Wards</a>
45 <a class="dropdown-item" href="Rooms.php">Rooms</a>
46
47 </div>
48 </li>
49 </ul>
```

Figure 3. 95 Coding for Patients.php part 1

```

44 </li>
45
46 <li class="nav-item dropdown">
47   <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-toggle="dropdown" > Supplies </a>
48
49   <div class="dropdown-menu">
50     <a class="dropdown-item" href="Medicines.php">Medicines</a>
51     <a class="dropdown-item" href="Other.php">Other</a>
52   </div>
53 </li>
54 </ul>
55
56   </div>
57
58 </div>
59
60 </nav>
61
62 <br><br><br>
63
64
65 <!-- Upper Part
-->
66
67 <div class="container">
68   <h1 class="text-center" >Patients Management</h1>
69
70   <a href="Functions_Patients/Create_Patient.php" class='btn btn-outline-dark mb-2'> <i class="bi bi-person-plus-fill"></i> Add New Patient</a>
71
72   <?php
73     echo "<a href='../Logout_Function.php?logout' class='btn btn-danger btn-outline-dark mb-2'> <i class='bi bi-box-arrow-in-left'></i> Logout</a>"<?
74   ?>
75
76 <!-- Table Part
-->
77
78 <table class="table table-striped table-bordered table-hover">
79   <thead class="table-dark">
80     <tr>
81       <th scope="col">ID</th>
82       <th scope="col">Name</th>
83       <th scope="col">Age</th>
84       <th scope="col">Ward</th>
85       <th scope="col" colspan="3" class="text-center">Options</th>
86     </tr>
87   </thead>
88
89   <tbody>
90     <tr>

```

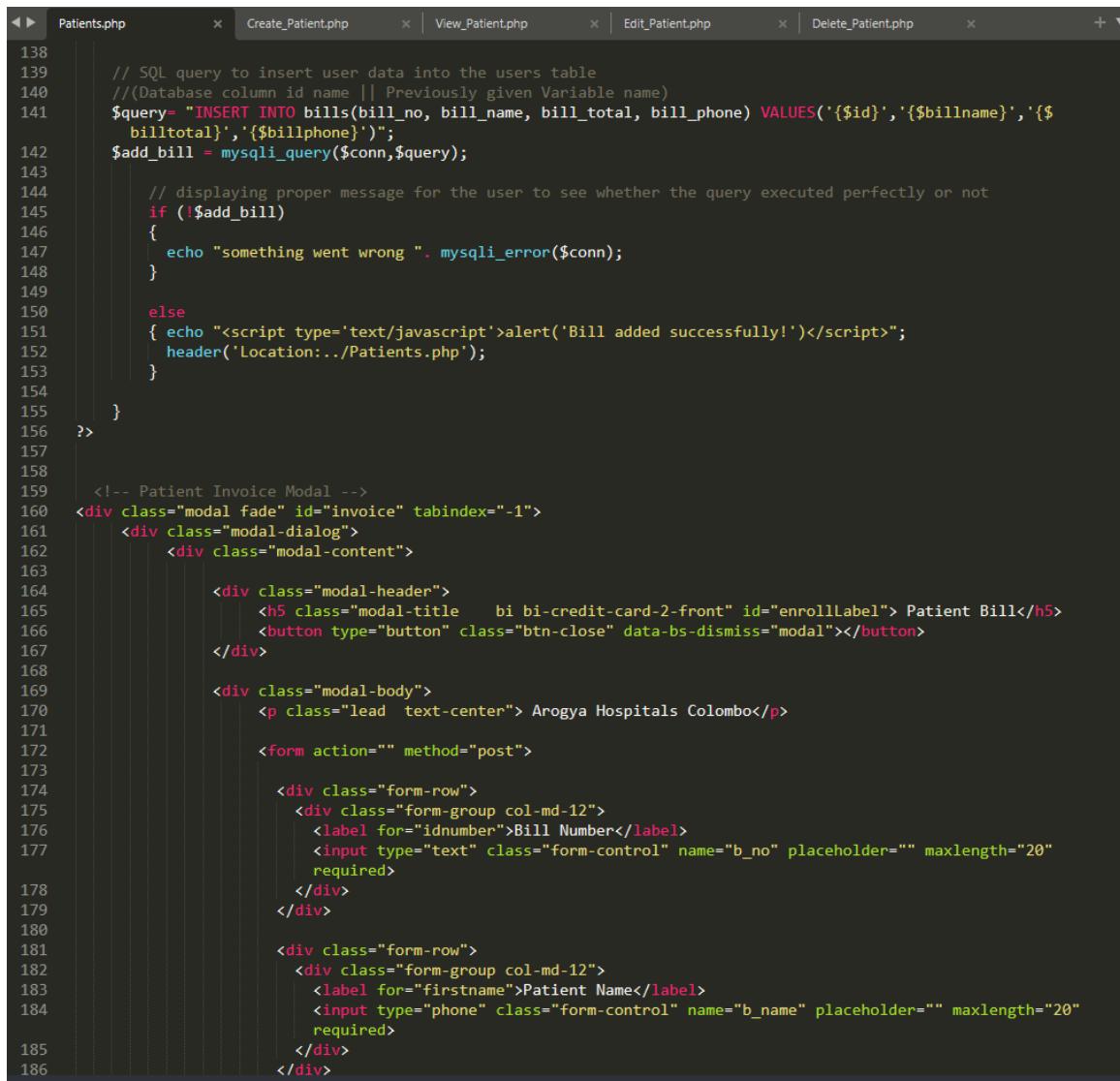
Figure 3. 96 Coding for Patients.php part 2

```

90     <tr>
91         <?php
92             $query="SELECT * FROM patient";           // SQL query to fetch all table data
93             $view_users= mysqli_query($conn,$query);    // sending the query to the database
94
95             // displaying all the data retrieved from the database using while loop
96             while($row= mysqli_fetch_assoc($view_users))
97             {
98                 //Now given Variable name || Database column id name
99                 $id = $row['id'];
100                $pidc = $row['p_card'];
101                $pfname = $row['p_f_name'];
102                $plname = $row['p_l_name'];
103                $page = $row['p_age'];
104                $pward = $row['p_admit'];
105
106                echo "<tr >";
107                echo " <th scope='row' >{$pidc}</th>";
108                echo " <td > {$pfname} {$plname}</td>";
109                echo " <td > {$page}</td>";
110                echo " <td >{$pward} </td>";
111
112                echo " <td class='text-center'> <a href='Functions_Patients/View_Patient.php?user_id={$id}'"
113                    class='btn btn-primary'> <i class='bi bi-eye'></i> View</a> </td>";
114                echo " <td class='text-center' > <a href='Functions_Patients/Edit_Patient.php?edit&user_id={$id}'"
115                    class='btn btn-secondary'><i class='bi bi-pencil'></i> Edit</a> </td>";
116
117                echo " <td class='text-center'> <a href='Functions_Patients/Delete_Patient.php?delete={$id}'"
118                    class='btn btn-danger'> <i class='bi bi-trash'></i> Delete</a> </td>";
119                echo " </tr >";
120            }
121        </tr>
122    </tbody>
123 </table>
124 </div>
125
126 <?php
127
128 //To give new values to DATABASE
129 if(isset($_POST['submit']))
130 {
131     //Now given Variable name || Textbox id name
132     $id = $_POST['b_no'];
133     $billname = $_POST['b_name'];
134     $billtotal = $_POST['b_total'];
135     $billphone = $_POST['b_phone'];
136
137
138

```

Figure 3. 97 Coding for Patients.php part 3

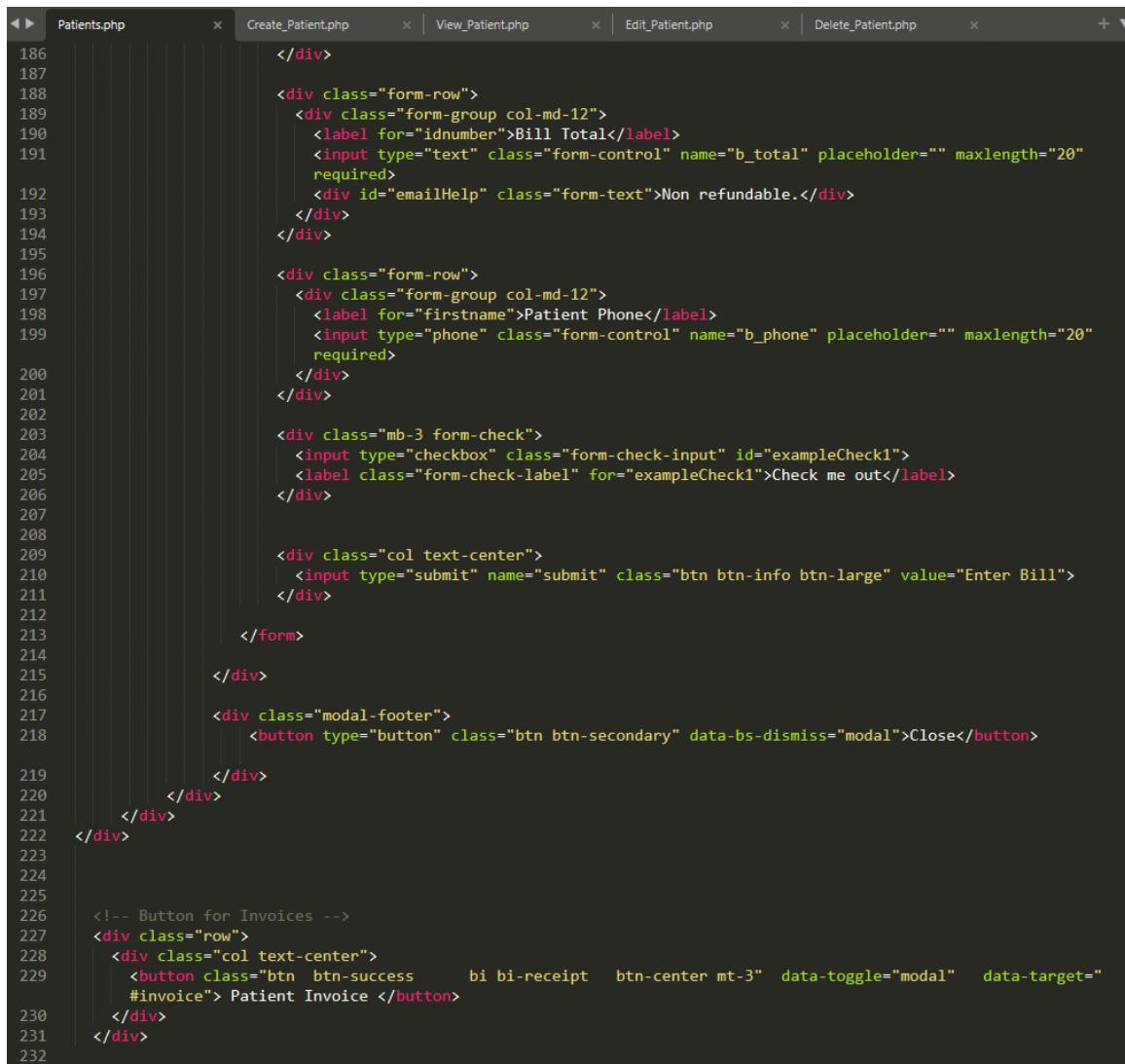


```

138     // SQL query to insert user data into the users table
139     // (Database column id name || Previously given Variable name)
140     $query= "INSERT INTO bills(bill_no, bill_name, bill_total, bill_phone) VALUES('{$id}', '{$billname}', '{$billtotal}', '{$billphone}')";
141     $add_bill = mysqli_query($conn,$query);
142
143     // displaying proper message for the user to see whether the query executed perfectly or not
144     if (!$add_bill)
145     {
146         echo "something went wrong ". mysqli_error($conn);
147     }
148
149     else
150     { echo "<script type='text/javascript'>alert('Bill added successfully!')</script>";
151       header('Location:../Patients.php');
152     }
153
154 }
155 ?>
156
157
158
159 <!-- Patient Invoice Modal -->
160 <div class="modal fade" id="invoice" tabindex="-1">
161   <div class="modal-dialog">
162     <div class="modal-content">
163
164       <div class="modal-header">
165         <h5 class="modal-title" bi bi-credit-card-2-front" id="enrollLabel"> Patient Bill</h5>
166         <button type="button" class="btn-close" data-bs-dismiss="modal"></button>
167       </div>
168
169       <div class="modal-body">
170         <p class="lead text-center"> Arogya Hospitals Colombo</p>
171
172         <form action="" method="post">
173
174           <div class="form-row">
175             <div class="form-group col-md-12">
176               <label for="idnumber">Bill Number</label>
177               <input type="text" class="form-control" name="b_no" placeholder="" maxlength="20"
178                   required>
179             </div>
180           </div>
181
182           <div class="form-row">
183             <div class="form-group col-md-12">
184               <label for="firstname">Patient Name</label>
185               <input type="text" class="form-control" name="b_name" placeholder="" maxlength="20"
186                   required>
187             </div>
188           </div>

```

Figure 3. 98 Coding for Patients.php part 4



```

186     </div>
187
188     <div class="form-row">
189         <div class="form-group col-md-12">
190             <label for="idnumber">Bill Total</label>
191             <input type="text" class="form-control" name="b_total" placeholder="" maxlength="20" required>
192             <div id="emailHelp" class="form-text">Non refundable.</div>
193         </div>
194     </div>
195
196     <div class="form-row">
197         <div class="form-group col-md-12">
198             <label for="firstname">Patient Phone</label>
199             <input type="phone" class="form-control" name="b_phone" placeholder="" maxlength="20" required>
200         </div>
201     </div>
202
203     <div class="mb-3 form-check">
204         <input type="checkbox" class="form-check-input" id="exampleCheck1">
205         <label class="form-check-label" for="exampleCheck1">Check me out</label>
206     </div>
207
208
209     <div class="col text-center">
210         <input type="submit" name="submit" class="btn btn-info btn-large" value="Enter Bill">
211     </div>
212
213 </form>
214
215 </div>
216
217 <div class="modal-footer">
218     <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button>
219
220 </div>
221 </div>
222 </div>
223
224
225
226 <!-- Button for Invoices -->
227 <div class="row">
228     <div class="col text-center">
229         <button class="btn btn-success bi bi-receipt btn-center mt-3" data-toggle="modal" data-target="#invoice"> Patient Invoice </button>
230     </div>
231 </div>
232

```

Figure 3. 99 Coding for Patients.php part 5

```
230     |     </div>
231     |     </div>
232
233     |     <!-- BACK button to go to the Home Page -->
234     |     <div class="container text-center mt-3">
235     |     |     <a href="Details_Pages/Home.php" class="btn btn-warning mt-5"> Back </a>
236     |     </div>
237
238
239
240
241     <script>
242     |     if ( window.history.replaceState ) {
243     |     |     window.history.replaceState( null, null, window.location.href );
244     |     }
245     </script>
```

Figure 3. 100 Coding for Patients.php part 6

Create_Patient.php :

The screenshot shows a web browser window with the title 'localhost/Arogya/Page_Files/Functions_Patients/Create_Patient.p...'. The main content is a form titled 'Add Patient Details' with the following fields:

- ID Number: Enter ID Number
- First Name: Enter First Name
- Last Name: Enter Last Name
- Mobile: Enter Mobile Number
- Address: Enter Address
- Gender: Choose...
- Age: Enter Age
- Admit To: Choose...
- Patient's Medical History: A large text area for input.

At the bottom are two buttons: a teal 'Submit' button and a yellow 'Back' button.

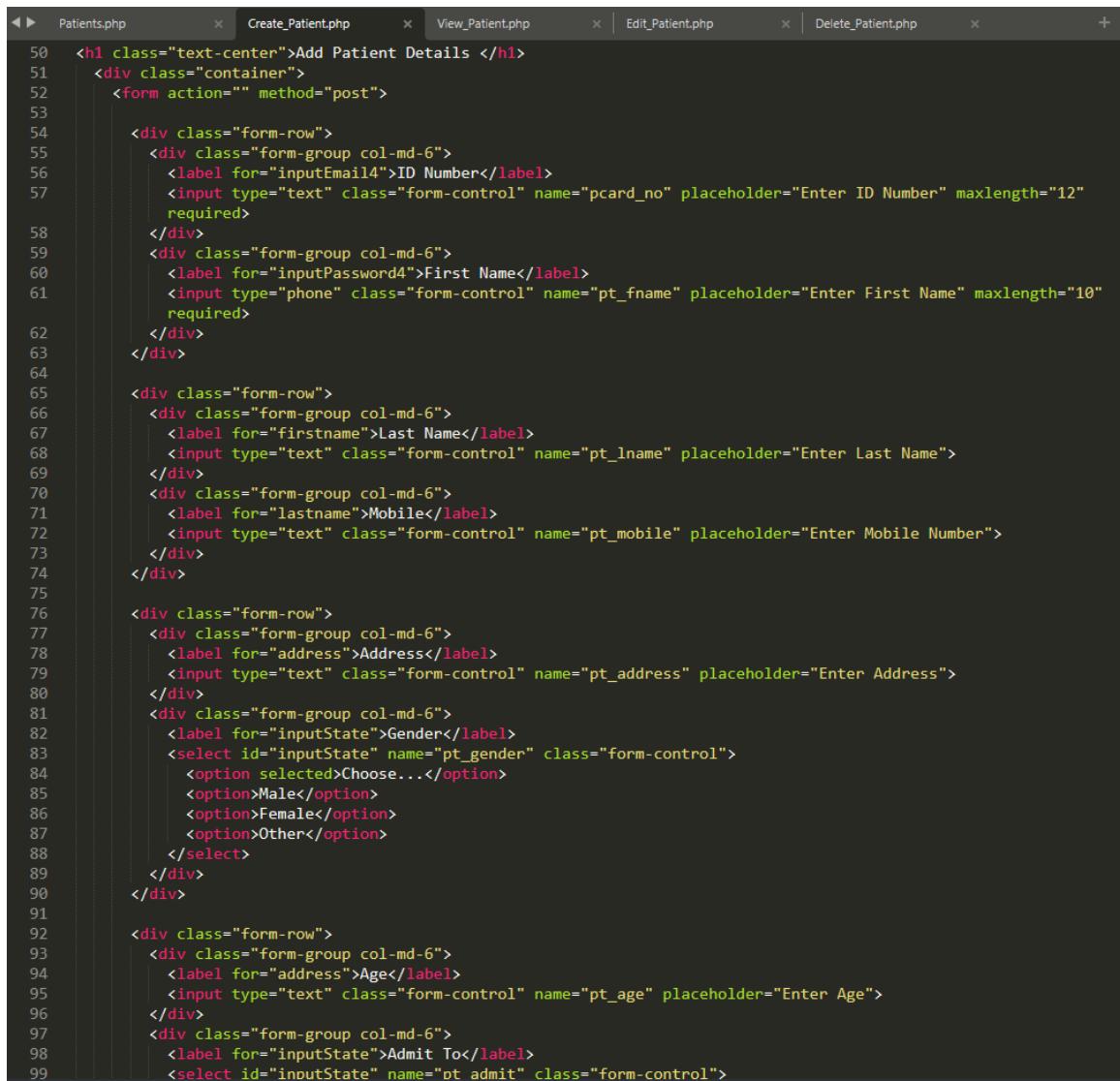
Figure 3. 101 Final Design of Create_Patient.php

```

1 <!-- Header -->
2 <?php include "../header.php" ?>
3
4 <?php
5
6 //To give new values to DATABASE
7 if(isset($_POST['submit']))
8 {
9     //((Now given Variable name || Textbox id name)
10    $id = $_POST['pcard_no'];
11    $pfname = $_POST['pt_fname'];
12    $plname = $_POST['pt_lname'];
13    $pphone = $_POST['pt_mobile'];
14    $paddress = $_POST['pt_address'];
15    $pgender = $_POST['pt_gender'];
16    $page = $_POST['pt_age'];
17    $padmit = $_POST['pt_admit'];
18    $phistory = $_POST['pt_history'];
19
20
21
22 // SQL query to insert user data into the users table
23 $query= "INSERT INTO patient(p_card, p_f_name, p_l_name, p_mobile, p_address, p_gender, p_age, p_admit,
24     p_history) VALUES('{$id}', '{$pfname}', '{$plname}', '{$pphone}', '{$paddress}', '{$pgender}', '{$page}', '{$
25     padmit}', '{$phistory}')";
26 $add_patient = mysqli_query($conn,$query);
27
28     // displaying proper message for the user to see whether the query executed perfectly or not
29     if (! $add_patient)
30     {
31         echo "something went wrong ". mysqli_error($conn);
32     }
33
34     else
35     { echo "<script type='text/javascript'>alert('User added successfully!')</script>";
36         header('Location: ../Patients.php');
37     }
38 }
39 <?php
40 //Select ward from ward list and give exact value to the database
41 $query = "SELECT w_number FROM wards";
42 $result_set = mysqli_query($conn, $query);
43 $ward_list = '';
44 while ($result = mysqli_fetch_assoc($result_set)) {
45     $ward_list .= "<option value=\"{$result['w_number']}\">{$result['w_number']}

```

Figure 3. 102 Coding for Create_Patient.php part I

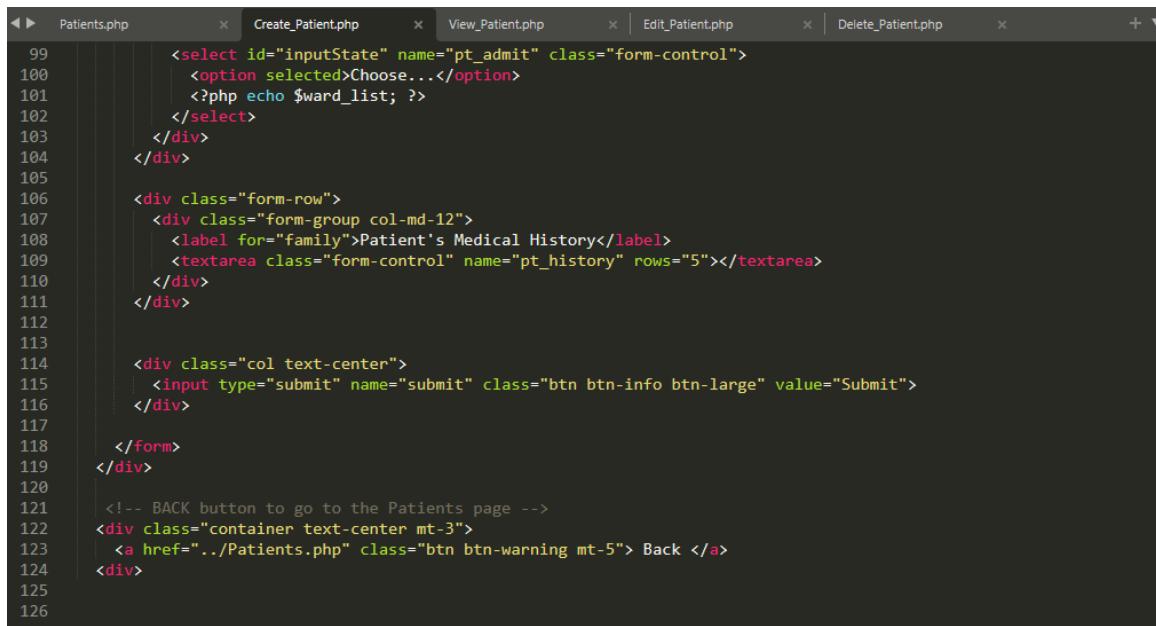


```

50   <h1 class="text-center">Add Patient Details </h1>
51   <div class="container">
52     <form action="" method="post">
53
54       <div class="form-row">
55         <div class="form-group col-md-6">
56           <label for="inputEmail4">ID Number</label>
57           <input type="text" class="form-control" name="pcard_no" placeholder="Enter ID Number" maxlength="12" required>
58         </div>
59         <div class="form-group col-md-6">
60           <label for="inputPassword4">First Name</label>
61           <input type="text" class="form-control" name="pt_fname" placeholder="Enter First Name" maxlength="10" required>
62         </div>
63     </div>
64
65     <div class="form-row">
66       <div class="form-group col-md-6">
67         <label for="firstname">Last Name</label>
68         <input type="text" class="form-control" name="pt_lname" placeholder="Enter Last Name">
69       </div>
70       <div class="form-group col-md-6">
71         <label for="lastname">Mobile</label>
72         <input type="text" class="form-control" name="pt_mobile" placeholder="Enter Mobile Number">
73       </div>
74     </div>
75
76     <div class="form-row">
77       <div class="form-group col-md-6">
78         <label for="address">Address</label>
79         <input type="text" class="form-control" name="pt_address" placeholder="Enter Address">
80       </div>
81       <div class="form-group col-md-6">
82         <label for="inputState">Gender</label>
83         <select id="inputState" name="pt_gender" class="form-control">
84           <option selected>Choose...</option>
85           <option>Male</option>
86           <option>Female</option>
87           <option>Other</option>
88         </select>
89       </div>
90     </div>
91
92     <div class="form-row">
93       <div class="form-group col-md-6">
94         <label for="address">Age</label>
95         <input type="text" class="form-control" name="pt_age" placeholder="Enter Age">
96       </div>
97       <div class="form-group col-md-6">
98         <label for="inputState">Admit To</label>
99         <select id="inputState" name="pt_admit" class="form-control">

```

Figure 3. 103 Coding for Create_Patient.php part 2



```
99     <select id="inputState" name="pt_admit" class="form-control">
100       <option selected>Choose...</option>
101       <?php echo $ward_list; ?>
102     </select>
103   </div>
104 </div>
105
106 <div class="form-row">
107   <div class="form-group col-md-12">
108     <label for="family">Patient's Medical History</label>
109     <textarea class="form-control" name="pt_history" rows="5"></textarea>
110   </div>
111 </div>
112
113 <div class="col text-center">
114   <input type="submit" name="submit" class="btn btn-info btn-large" value="Submit">
115 </div>
116
117 </form>
118 </div>
119
120
121 <!-- BACK button to go to the Patients page -->
122 <div class="container text-center mt-3">
123   <a href="../../Patients.php" class="btn btn-warning mt-5"> Back </a>
124 <div>
125
126
```

Figure 3. 104 Coding for Create_Patient.php part 3

View_Patient.php :

The screenshot shows a web browser window with the URL `localhost/Arogya/Page_Files/Functions_Patients/View_Patient.php...` in the address bar. The main content is titled "Patient Details". A table displays patient information:

| ID | Name | Mobile | Address | Gender | Age | Admitted to | Patient History |
|----|-----------------|------------|---------|--------|-----|-------------|-----------------------|
| 1 | Judith Michelle | 0764170485 | test | Male | 20 | Ward 2 | Good medical history. |

At the bottom right of the page is a yellow button labeled "Back".

Figure 3. 105 Final Design of View_Patient.php

```

1 <!-- Header -->
2 <?php include '../header.php'>
3
4 <h1 class="text-center">Patient Details</h1>
5
6 <div class="container">
7
8     <table class="table table-striped table-bordered table-hover">
9         <thead class="table-dark">
10            <tr>
11                <th scope="col">ID</th>
12                <th scope="col">Name</th>
13                <th scope="col">Mobile</th>
14                <th scope="col">Address</th>
15                <th scope="col">Gender</th>
16                <th scope="col">Age</th>
17                <th scope="col">Admitted to</th>
18                <th scope="col">Patient History</th>
19            </tr>
20        </thead>
21        <tbody>
22            <tr>
23                <?php
24                    // first we check using 'isset()' function if the variable is set or not
25                    //Processing form data when form is submitted
26                    if (isset($_GET['user_id']))
27                {
28                        $userid = $_GET['user_id'];
29
30                        // SQL query to fetch the data where id=$userid & storing data in view_user
31                        $query="SELECT * FROM patient WHERE id = {$userid} ";
32                        $view_staff= mysqli_query($conn,$query);
33
34                        while($row = mysqli_fetch_assoc($view_staff))
35                        {
36                            $id = $row['id'];
37                            $pidc = $row['p_card'];
38                            $pfName = $row['p_f_name'];
39                            $plName = $row['p_l_name'];
40                            $pMobile = $row['p_mobile'];
41                            $pAddress = $row['p_address'];
42                            $pGender = $row['p_gender'];
43                            $page = $row['p_age'];
44                            $pward = $row['p_admit'];
45                            $phistory = $row['p_history'];
46
47                            echo "<tr >";
48                            echo " <td >{$pidc}</td>";
49                            echo " <td > {$pfName} {$plName} </td>";
50                            echo " <td > {$pMobile}</td>";
51                            echo " <td >{$oAddress} </td>";
52

```

Figure 3. 106 Coding for View_Patient.php part 1

```
49     echo " <td >{$pidc}</td>";
50     echo " <td >{$pfName} {$plName} </td>";
51     echo " <td >{$pMobile}</td>";
52     echo " <td >{$pAddress} </td>";
53     echo " <td >{$pGender}</td>";
54     echo " <td >{$page} </td>";
55     echo " <td >{$pward} </td>";
56     echo " <td >{$phistory} </td>";
57     echo " </tr> ";
58 }
59     ?
60     </tr>
61     </tbody>
62   </table>
63 </div>
64
65     <!-- BACK Button to go to previous page --&gt;
66 &lt;div class="container text-center mt-5"&gt;
67   &lt;a href="../patients.php" class="btn btn-warning mt-5"&gt; Back &lt;/a&gt;
68 &lt;/div&gt;
69
70
71     <!-- Footer --&gt;
72 &lt;?php include "../footer.php" ?&gt;</pre>
```

Figure 3. 107 Coding for View_Patient.php part 2

Edit_Patient.php :

The screenshot shows a web browser window with the URL `localhost/Arogya/Page_Files/Functions_Patients/Edit_Patient.php...`. The page title is "Update Patient Details". The form contains the following fields:

| | |
|---------------------------|------------|
| ID Number | First Name |
| 1 | Judith |
| Last Name | Mobile |
| Michelle | 0764170485 |
| Address | Gender |
| test | Male |
| Age | Admit To |
| 20 | Ward 2 |
| Patient's Medical History | |
| Good medical history. | |

At the bottom right of the form area is a blue "Submit" button, and at the bottom center is a yellow "Back" button.

Figure 3. 108 Final Design of Edit_Patient.php

```

1  <!-- Header -->
2  <?php include "../../header.php"?>
3
4  <?php
5      // checking if the variable is set or not and if set adding the set data value to variable userid
6      if(isset($_GET['user_id']))
7      {
8          $userid = $_GET['user_id'];
9      }
10     // SQL query to select all the data from the table where id = $userid
11     $query="SELECT * FROM patient WHERE id = $userid ";
12     $edit_staff= mysqli_query($conn,$query);
13
14     while($row = mysqli_fetch_assoc($edit_staff))
15     {
16         //Textbox echo id name || Database column id name
17         $id = $row['id'];
18         $ptidc = $row['p_card'];
19         $ptfName = $row['p_f_name'];
20         $ptlName = $row['p_l_name'];
21         $ptMobile = $row['p_mobile'];
22         $ptAddress = $row['p_address'];
23         $ptGender = $row['p_gender'];
24         $ptage = $row['p_age'];
25         $ptward = $row['p_admit'];
26         $pthistory = $row['p_history'];
27     }
28
29     //Processing form data when form is submitted
30     if(isset($_POST['update']))
31     {
32         //Now given Variable name || Textbox id name
33         $ptidc = $_POST['pcard_no'];
34         $ptfName = $_POST['pt_fname'];
35         $ptlName = $_POST['pt_lname'];
36         $ptMobile = $_POST['pt_mobile'];
37         $ptAddress = $_POST['pt_address'];
38         $ptGender = $_POST['pt_gender'];
39         $ptage = $_POST['pt_age'];
40         $ptward = $_POST['pt_admit'];
41         $pthistory = $_POST['pt_history'];
42
43         // SQL query to update the data in user table where the id = $userid
44         //(Database column id name || Previously given Variable name)
45         $query = "UPDATE patient SET p_card = '{$ptidc}' , p_f_name = '{$ptfName}' , p_l_name = '{$ptlName}' ,
46             p_mobile = '{$ptMobile}' , p_address = '{$ptAddress}' , p_gender = '{$ptGender}' , p_age = '{$ptage}' ,
47             p_admit = '{$ptward}' , p_history = '{$pthistory}' WHERE id = $userid";
48         $update_user = mysqli_query($conn, $query);
49         echo "<script type='text/javascript'>alert('Patient edited successfully!')</script>";
50         header("location: ../../patients.php");
51     }
52 ?>

```

Figure 3. 109 Coding for Edit_Patient.php part 1

```

50    ?>
51
52
53 <?php
54     //Select ward from ward list and give exact value to the database
55     $query = "SELECT w_number FROM wards";
56     $result_set = mysqli_query($conn, $query);
57     $ward_list = '';
58     while ($result = mysqli_fetch_assoc($result_set))
59     {
60         $ward_list .= "<option value=\"{$result['w_number']}\" {$result['w_number']}></option>";
61     }
62
63 ?>
64
65
66 <h1 class="text-center">Update Patient Details</h1>
67 <div class="container">
68     <form action="" method="post">
69
70         <div class="form-row">
71             <div class="form-group col-md-6">
72                 <label for="inputEmail4">ID Number</label>
73                 <input type="text" class="form-control" name="pcard_no" value="<?php echo $pidc ?>" maxlength="12"
74                     required>
75             </div>
76             <div class="form-group col-md-6">
77                 <label for="inputPassword4">First Name</label>
78                 <input type="phone" class="form-control" name="pt_fname" value="<?php echo $pfName ?>" maxlength="10"
79                     required>
80             </div>
81         </div>
82         <div class="form-row">
83             <div class="form-group col-md-6">
84                 <label for="firstname">Last Name</label>
85                 <input type="text" class="form-control" name="pt_lname" value="<?php echo $plName ?>">
86             </div>
87             <div class="form-group col-md-6">
88                 <label for="lastname">Mobile</label>
89                 <input type="text" class="form-control" name="pt_mobile" value="<?php echo $pMobile ?>">
90             </div>
91         </div>
92         <div class="form-row">
93             <div class="form-group col-md-6">
94                 <label for="address">Address</label>
95                 <input type="text" class="form-control" name="pt_address" value="<?php echo $pAddress ?>">
96             </div>
97             <div class="form-group col-md-6">
98                 <label for="inputState">Gender</label>
99                 <select id="inputState" name="pt_gender" class="form-control" value="<?php echo $pGender ?>">
```

Figure 3. 110 Coding for Edit_Patient.php part 2

```
96     </div>
97     <div class="form-group col-md-6">
98         <label for="inputState">Gender</label>
99         <select id="inputState" name="pt_gender" class="form-control" value=<?php echo $pGender ?>>
100            <option selected><?php echo $pGender ?></option>
101            <option>Male</option>
102            <option>Female</option>
103            <option>Other</option>
104        </select>
105    </div>
106 </div>
107
108 <div class="form-row">
109     <div class="form-group col-md-6">
110         <label for="address">Age</label>
111         <input type="text" class="form-control" name="pt_age" value=<?php echo $page ?>>
112     </div>
113     <div class="form-group col-md-6">
114         <label for="inputState">Admit To</label>
115         <select id="inputState" name="pt_admit" class="form-control" value=<?php echo $pward ?>>
116            <option selected><?php echo $pward ?></option>
117            <?php echo $ward_list; ?>
118        </select>
119    </div>
120 </div>
121
122 <div class="form-row">
123     <div class="form-group col-md-12">
124         <label for="family">Patient's Medical History</label>
125         <textarea class="form-control" name="pt_history"><?php echo htmlspecialchars($phistory); ?></textarea>
126     </div>
127 </div>
128
129 <div class="col text-center">
130     <input type="submit" name="update" class="btn btn-info btn-large" value="Submit">
131 </div>
132
133 </form>
134 </div>
135
136 <!-- BACK button to go to the Patients page -->
137 <div class="container text-center mt-5">
138     <a href="../Patients.php" class="btn btn-warning mt-5"> Back </a>
139 <div>
```

Figure 3. 111 Coding for Edit_Patient.php part 3

Delete_Patient.php :

```
1 | <!-- Footer -->
2 | <?php include "../header.php" ?>
3 |
4 | <?php
5 |     if(isset($_GET['delete']))
6 |     {
7 |         $userid= $_GET['delete'];
8 |
9 |         // SQL query to delete data from user table where id = $userid
10 |        $query = "DELETE FROM patient WHERE id = {$userid}";
11 |        $delete_query= mysqli_query($conn, $query);
12 |        header("Location: ../Patients.php");
13 |    }
14 |?>
15 |
16 |
17 |
```

Figure 3. 112 Coding for Delete_Patient.php part 1

Ward Management Page

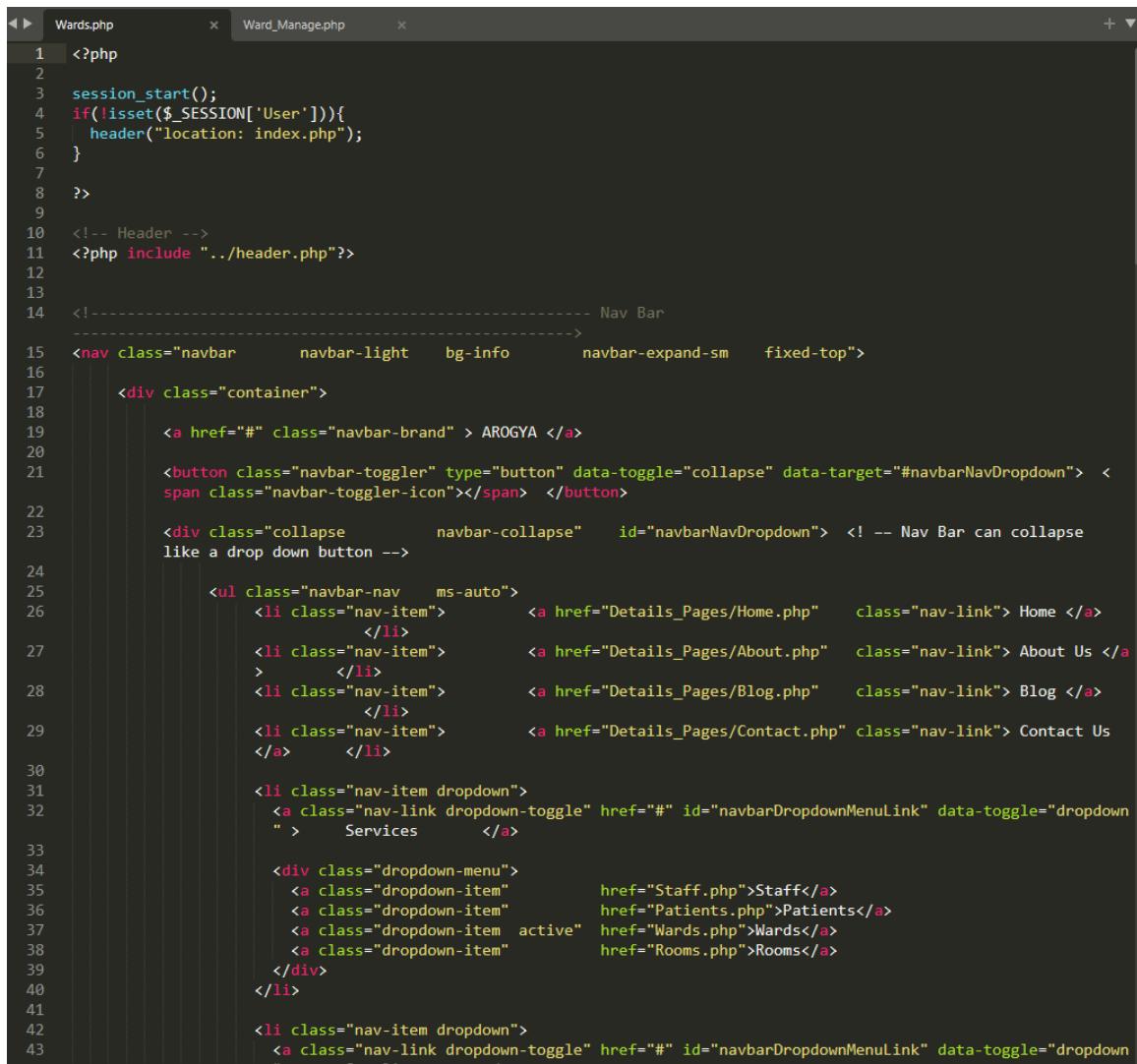
Wards.php :

The screenshot shows a web application interface titled "Ward Management". At the top, there is a navigation bar with links for Home, About Us, Blog, Contact Us, Services, and Supplies. A "Logout" button is also present. The main content area is titled "Ward Management". It features a table with four rows, each representing a ward (Ward 1, Ward 2, Ward 3, Ward 4). Each row has a "Manage the Ward" button. Below this is another table titled "Ward Details" with columns for Ward Number, Doctor 1, Doctor 2, Nurse 1, Nurse 2, Attendant 1, and Attendant 2. The data for each ward is listed as follows:

| Ward Number | Doctor 1 | Doctor 2 | Nurse 1 | Nurse 2 | Attendant 1 | Attendant 2 |
|-------------|----------------|----------------|--------------------|--------------|--------------------|--------------|
| Ward 1 | Pavel Fernando | Sadew Fernando | Ryan Wickramaratne | Mashi Perera | Kyle Wickramaratne | Shyni Perera |
| Ward 2 | Pavel Fernando | | Ryan Wickramaratne | | Kyle Wickramaratne | |
| Ward 3 | Sadew Fernando | | Mashi Perera | | Shyni Perera | |
| Ward 4 | Pavel Fernando | | Mashi Perera | | | |

A "Back" button is located at the bottom right of the page.

Figure 3. 113 Final Design of Wards.php



```

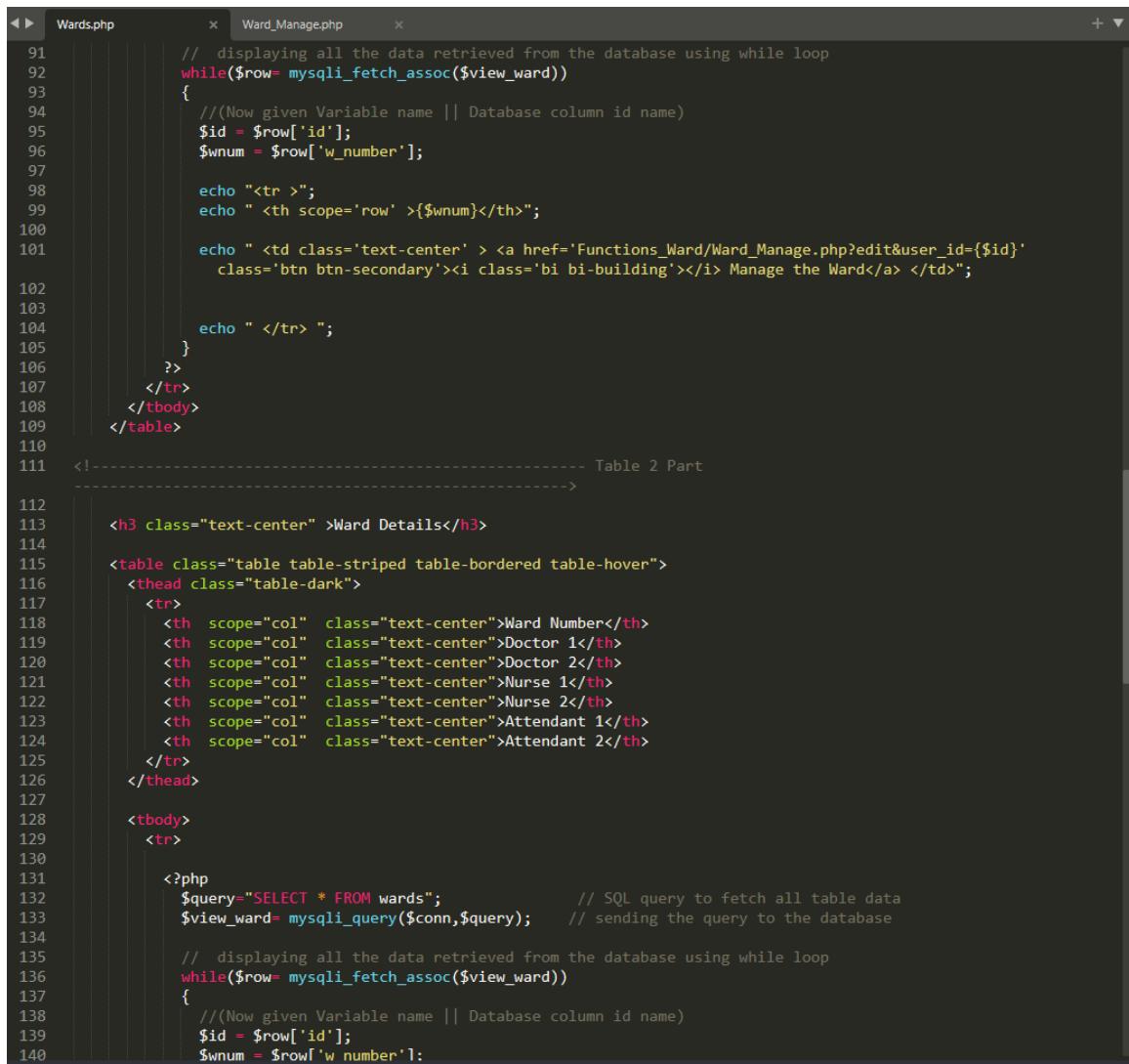
1 <?php
2
3 session_start();
4 if(!isset($_SESSION['User'])){
5 | header("location: index.php");
6 }
7
8 ?>
9
10 <!-- Header -->
11 <?php include "../header.php"?>
12
13
14 <!-- Nav Bar -->
15 <nav class="navbar navbar-light bg-info navbar-expand-sm fixed-top">
16
17 <div class="container">
18
19 <a href="#" class="navbar-brand" > AROGYA </a>
20
21 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavDropdown" >
22 <span class="navbar-toggler-icon"></span> </button>
23
24 <div class="collapse navbar-collapse" id="navbarNavDropdown" > <!-- Nav Bar can collapse
like a drop down button -->
25
26 <ul class="navbar-nav ms-auto">
27 <li class="nav-item"> <a href="Details_Pages/Home.php" class="nav-link" > Home </a>
28 </li>
29 <li class="nav-item"> <a href="Details_Pages/About.php" class="nav-link" > About Us </a>
30 </li>
31 <li class="nav-item"> <a href="Details_Pages/Blog.php" class="nav-link" > Blog </a>
32 </li>
33 <li class="nav-item"> <a href="Details_Pages/Contact.php" class="nav-link" > Contact Us
</a> </li>
34
35 <li class="nav-item dropdown">
36 <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-toggle="dropdown"
" > Services </a>
37
38 <div class="dropdown-menu">
39 <a class="dropdown-item" href="Staff.php" >Staff</a>
40 <a class="dropdown-item" href="Patients.php" >Patients</a>
41 <a class="dropdown-item active" href="Wards.php" >Wards</a>
42 <a class="dropdown-item" href="Rooms.php" >Rooms</a>
43 </div>

```

Figure 3. 114 Coding for Wards.php part I

```
Wards.php      Ward_Manage.php
44             " > Supplies </a>
45             <div class="dropdown-menu">
46                 <a class="dropdown-item" href="Medicines.php">Medicines</a>
47                 <a class="dropdown-item" href="Other.php">Other</a>
48             </div>
49         </li>
50     </ul>
51 
52     </div>
53 
54 </div>
55 
56 </nav>
57 
58 <br><br><br>
59 
60 
61 
62 
63 <!-- Upper Part
-->
64 
65 <div class="container">
66     <h1 class="text-center" >Ward Management</h1>
67 
68     <?php
69         echo "<a href='../Logout_Function.php?logout' class='btn btn-danger btn-outline-dark mb-2'> <i class='bi bi-box-arrow-in-left'></i> Logout</a>" ;
70     ?>
71 
72 
73 
74 <!-- Table 1 Part
-->
75 
76 <table class="table table-striped table-bordered table-hover">
77     <thead class="table-dark">
78         <tr>
79             <th scope="col">Ward</th>
80             <th scope="col" colspan="3" class="text-center">Edit Ward Status</th>
81         </tr>
82     </thead>
83 
84     <tbody>
85         <tr>
86 
87             <?php
88                 $query="SELECT * FROM wards";           // SQL query to fetch all table data
89                 $view_ward= mysqli_query($conn,$query); // sending the query to the database
90 
91             // displaying all the data retrieved from the database using while loop
92         </tr>
93     </tbody>
94 </table>
```

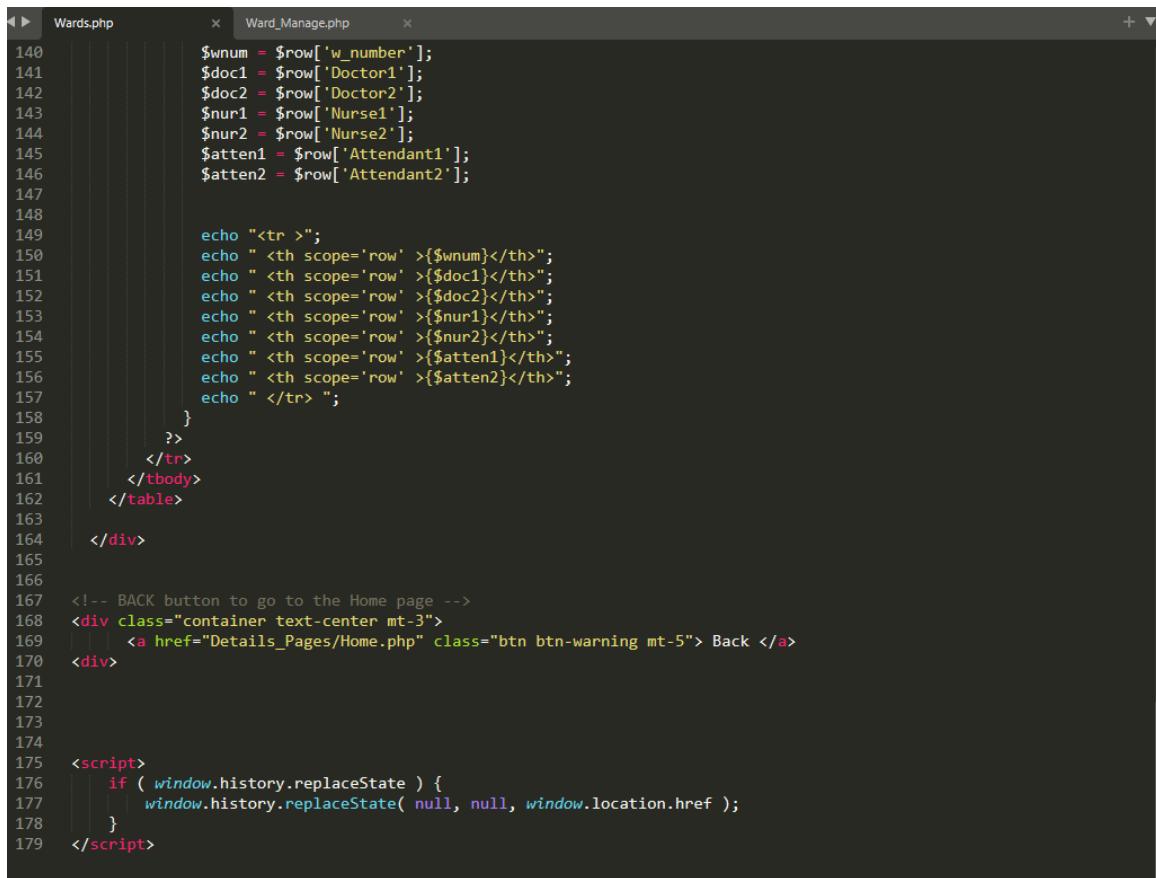
Figure 3. 115 Coding for Wards.php part 2



The screenshot shows a code editor with two tabs: "Wards.php" and "Ward_Manage.php". The "Wards.php" tab is active and displays the following PHP code:

```
91 // displaying all the data retrieved from the database using while loop
92 while($row= mysqli_fetch_assoc($view_ward))
93 {
94 //((Now given Variable name || Database column id name)
95 $id = $row['id'];
96 $wnum = $row['w_number'];
97
98 echo "<tr >";
99 echo " <th scope='row' >{$wnum}</th>";
100
101 echo " <td class='text-center' > <a href='Functions_Ward/Ward_Manage.php?edit&user_id={$id}'>
102 class='btn btn-secondary'><i class='bi bi-building'></i> Manage the Ward</a> </td>"; 
103
104 echo " </tr > ";
105 }
106 ?>
107 </tr>
108 </tbody>
109 </table>
110
111 <!----- Table 2 Part ----->
112
113 <h3 class="text-center" >Ward Details</h3>
114
115 <table class="table table-striped table-bordered table-hover">
116 <thead class="table-dark">
117 <tr>
118 <th scope="col" class="text-center">Ward Number</th>
119 <th scope="col" class="text-center">Doctor 1</th>
120 <th scope="col" class="text-center">Doctor 2</th>
121 <th scope="col" class="text-center">Nurse 1</th>
122 <th scope="col" class="text-center">Nurse 2</th>
123 <th scope="col" class="text-center">Attendant 1</th>
124 <th scope="col" class="text-center">Attendant 2</th>
125 </tr>
126 </thead>
127
128 <tbody>
129 <tr>
130
131 <?php
132 $query="SELECT * FROM wards";           // SQL query to fetch all table data
133 $view_ward= mysqli_query($conn,$query); // sending the query to the database
134
135 // displaying all the data retrieved from the database using while loop
136 while($row= mysqli_fetch_assoc($view_ward))
137 {
138 //((Now given Variable name || Database column id name)
139 $id = $row['id'];
140 $wnum = $row['w_number'];
```

Figure 3. 116 Coding for Wards.php part 3



The screenshot shows a code editor window with two tabs: "Wards.php" and "Ward_Manage.php". The "Wards.php" tab is active, displaying the following PHP code:

```
140     $wnum = $row['w_number'];
141     $doc1 = $row['Doctor1'];
142     $doc2 = $row['Doctor2'];
143     $nur1 = $row['Nurse1'];
144     $nur2 = $row['Nurse2'];
145     $atten1 = $row['Attendant1'];
146     $atten2 = $row['Attendant2'];
147
148     echo "<tr >";
149     echo " <th scope='row' >{$wnum}</th>";
150     echo " <th scope='row' >{$doc1}</th>";
151     echo " <th scope='row' >{$doc2}</th>";
152     echo " <th scope='row' >{$nur1}</th>";
153     echo " <th scope='row' >{$nur2}</th>";
154     echo " <th scope='row' >{$atten1}</th>";
155     echo " <th scope='row' >{$atten2}</th>";
156     echo "</tr> ";
157 }
158 ?>
159 </tr>
160 </tbody>
161 </table>
162
163 </div>
164
165
166
167 <!-- BACK button to go to the Home page -->
168 <div class="container text-center mt-3">
169   <a href="Details_Pages/Home.php" class="btn btn-warning mt-5"> Back </a>
170 </div>
171
172
173
174
175 <script>
176   if ( window.history.replaceState ) {
177     | | window.history.replaceState( null, null, window.location.href );
178   }
179 </script>
```

Figure 3. 117 Coding for Wards.php part 4

Ward_Manage.php :

The screenshot shows a web page titled "Ward 1". The page is divided into three sections: "Doctor", "Nurse", and "Attendant". Each section contains two input fields for names, labeled "(1)" and "(2)". The "Doctor" section has fields for "Pavel Fernandoooo" and "Sadew Fernando". The "Nurse" section has fields for "Ryan Wickramaratne" and "Mashi Perera". The "Attendant" section has fields for "Kyle Wickramaratne" and "Shyni Perera". A "Submit" button is located below the attendant section, and a "Back" button is at the bottom center.

| Section | Name (1) | Name (2) |
|-----------|--------------------|----------------|
| Doctor | Pavel Fernandoooo | Sadew Fernando |
| Nurse | Ryan Wickramaratne | Mashi Perera |
| Attendant | Kyle Wickramaratne | Shyni Perera |

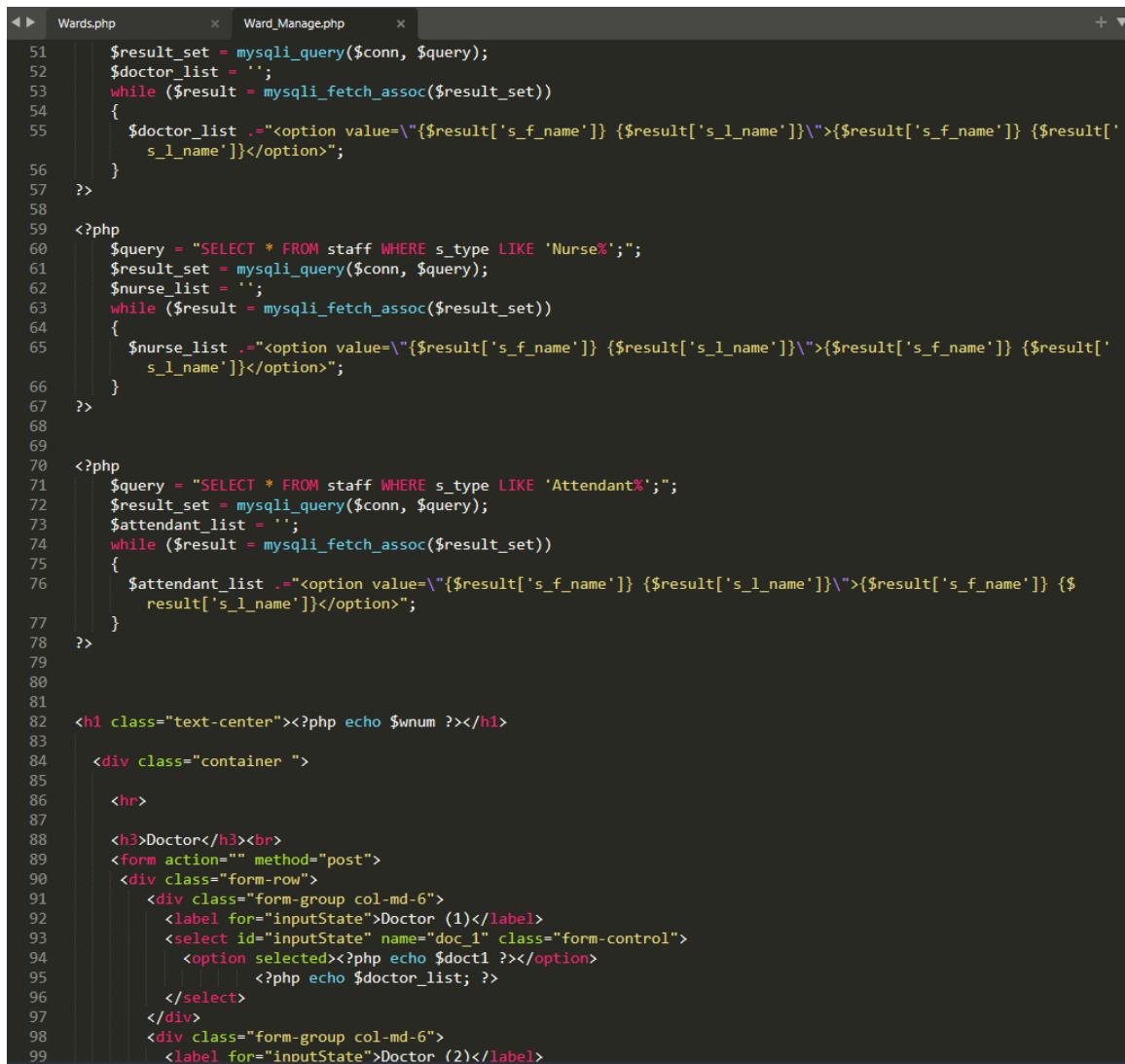
Figure 3. 118 Final Design of Ward_Manage.php

```

1 <!-- Footer -->
2 <?php include "../header.php"?>
3
4 <?php
5 // checking if the variable is set or not and if set adding the set data value to variable userid
6 if(isset($_GET['user_id']))
7 {
8     $userid = $_GET['user_id'];
9 }
10 // SQL query to select all the data from the table where id = $userid
11 $query="SELECT * FROM wards WHERE id = $userid ";
12 $ward_manage= mysqli_query($conn,$query);
13
14 while($row = mysqli_fetch_assoc($ward_manage))
15 {
16     //((Textbox echo id name || Database column id name)
17     $id = $row['id'];
18     $wnum = $row['w_number'];
19     $doct1 = $row['Doctor1'];
20     $doct2 = $row['Doctor2'];
21     $nurs1 = $row['Nurse1'];
22     $nurs2 = $row['Nurse2'];
23     $atten1 = $row['Attendant1'];
24     $atten2 = $row['Attendant2'];
25
26 }
27
28 //Processing form data when form is submitted
29 if(isset($_POST['update']))
30 {
31     //((Now given Variable name || Textbox id name)
32     $doc1 = $_POST['doc_1'];
33     $doc2 = $_POST['doc_2'];
34     $nur1 = $_POST['nur_1'];
35     $nur2 = $_POST['nur_2'];
36     $atte1 = $_POST['atte_1'];
37     $atte2 = $_POST['atte_2'];
38
39
40     // SQL query to update the data in user table where the id = $userid
41     //(Database column id name || Previously given Variable name)
42     $query = "UPDATE wards SET Doctor1 = '{$doc1}' , Doctor2 = '{$doc2}' , Nurse1 = '{$nur1}' , Nurse2 = '{$nur2}' ,
43     , Attendant1 = '{$atte1}' , Attendant2 = '{$atte2}' WHERE id = $userid";
44     $update_user = mysqli_query($conn, $query);
45     echo "<script type='text/javascript'>alert('Ward data updated successfully!')</script>";
46     header("location:../Wards.php");
47 }
48
49 <?php
50     $query = "SELECT * FROM staff WHERE s_type LIKE 'Doctor%'";
51     $result_set = mvsali auerv($conn, $auerv);

```

Figure 3. 119 Coding for Ward_Manage.php part I

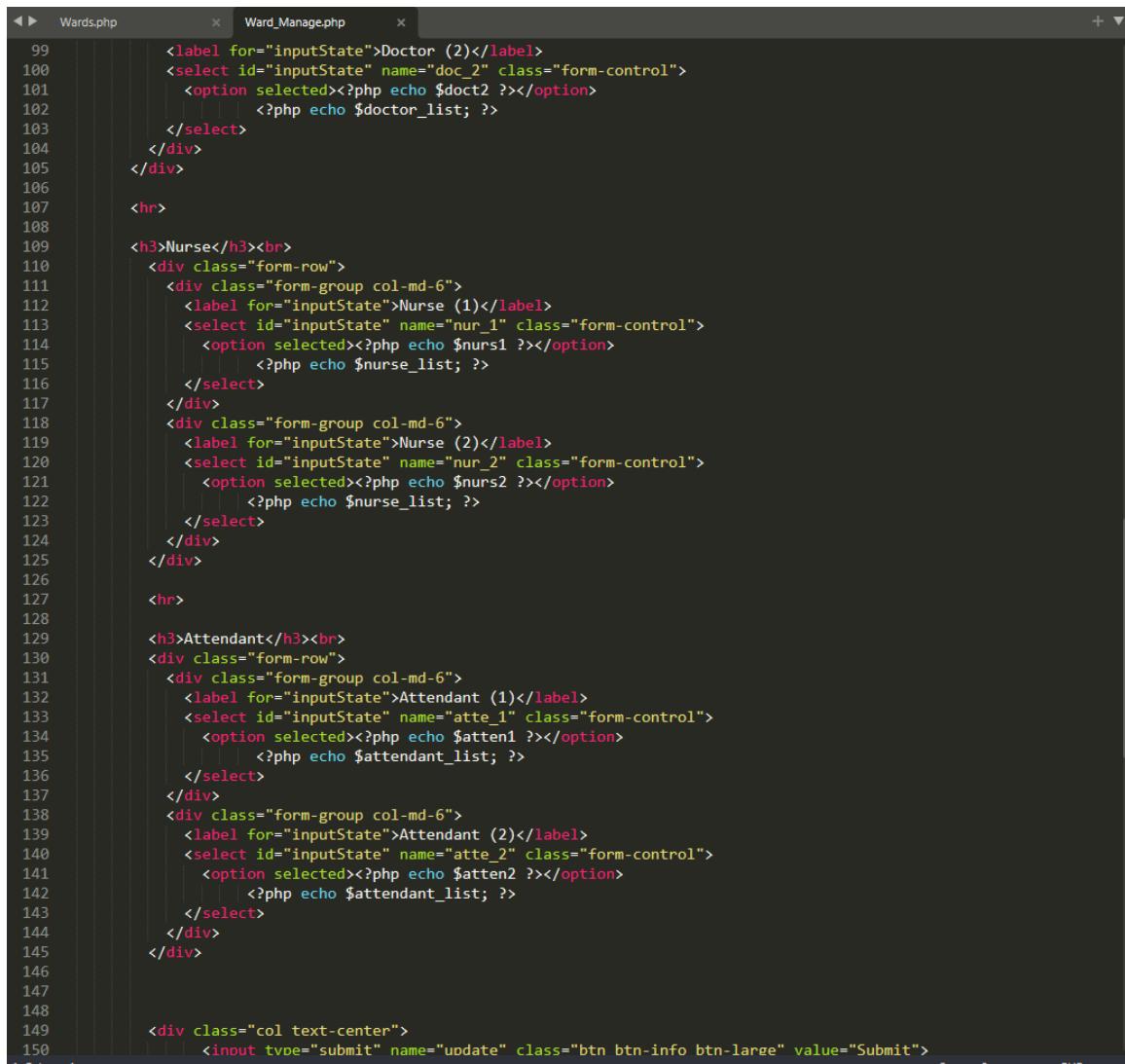


```

51     $result_set = mysqli_query($conn, $query);
52     $doctor_list = '';
53     while ($result = mysqli_fetch_assoc($result_set))
54     {
55         $doctor_list .= "<option value=\"$result['s_f_name']\" {$result['s_l_name']}>{$result['s_f_name']} {$result['
56             s_l_name']}

```

Figure 3. 120 Coding for Ward_Manage.php part 2



```

99         <label for="inputState">Doctor (2)</label>
100        <select id="inputState" name="doc_2" class="form-control">
101          <option selected><?php echo $doct2 ?></option>
102          <?php echo $doctor_list; ?>
103        </select>
104      </div>
105    </div>
106
107    <hr>
108
109    <h3>Nurse</h3><br>
110    <div class="form-row">
111      <div class="form-group col-md-6">
112        <label for="inputState">Nurse (1)</label>
113        <select id="inputState" name="nur_1" class="form-control">
114          <option selected><?php echo $nurs1 ?></option>
115          <?php echo $nurse_list; ?>
116        </select>
117      </div>
118      <div class="form-group col-md-6">
119        <label for="inputState">Nurse (2)</label>
120        <select id="inputState" name="nur_2" class="form-control">
121          <option selected><?php echo $nurs2 ?></option>
122          <?php echo $nurse_list; ?>
123        </select>
124      </div>
125    </div>
126
127    <hr>
128
129    <h3>Attendant</h3><br>
130    <div class="form-row">
131      <div class="form-group col-md-6">
132        <label for="inputState">Attendant (1)</label>
133        <select id="inputState" name="atte_1" class="form-control">
134          <option selected><?php echo $atten1 ?></option>
135          <?php echo $attendant_list; ?>
136        </select>
137      </div>
138      <div class="form-group col-md-6">
139        <label for="inputState">Attendant (2)</label>
140        <select id="inputState" name="atte_2" class="form-control">
141          <option selected><?php echo $atten2 ?></option>
142          <?php echo $attendant_list; ?>
143        </select>
144      </div>
145    </div>
146
147
148
149    <div class="col text-center">
150      <input type="submit" name="update" class="btn btn-info btn-large" value="Submit">

```

Figure 3. 121 Coding for Ward_Manage.php part 3

```
150     <input type="submit" name="update" class="btn btn-info btn-large" value="Submit">
151   </div>
152
153   </form>
154 </div>
155
156  <!-- aBACK button to go to the Wards page -->
157  <div class="container text-center mt-5">
158    | <a href="../Wards.php" class="btn btn-warning mt-5"> Back </a>
159  <div>
160
161
162
```

Figure 3. 122 Coding for Ward_Manage.php part 4

Room Management Page

Rooms.php :

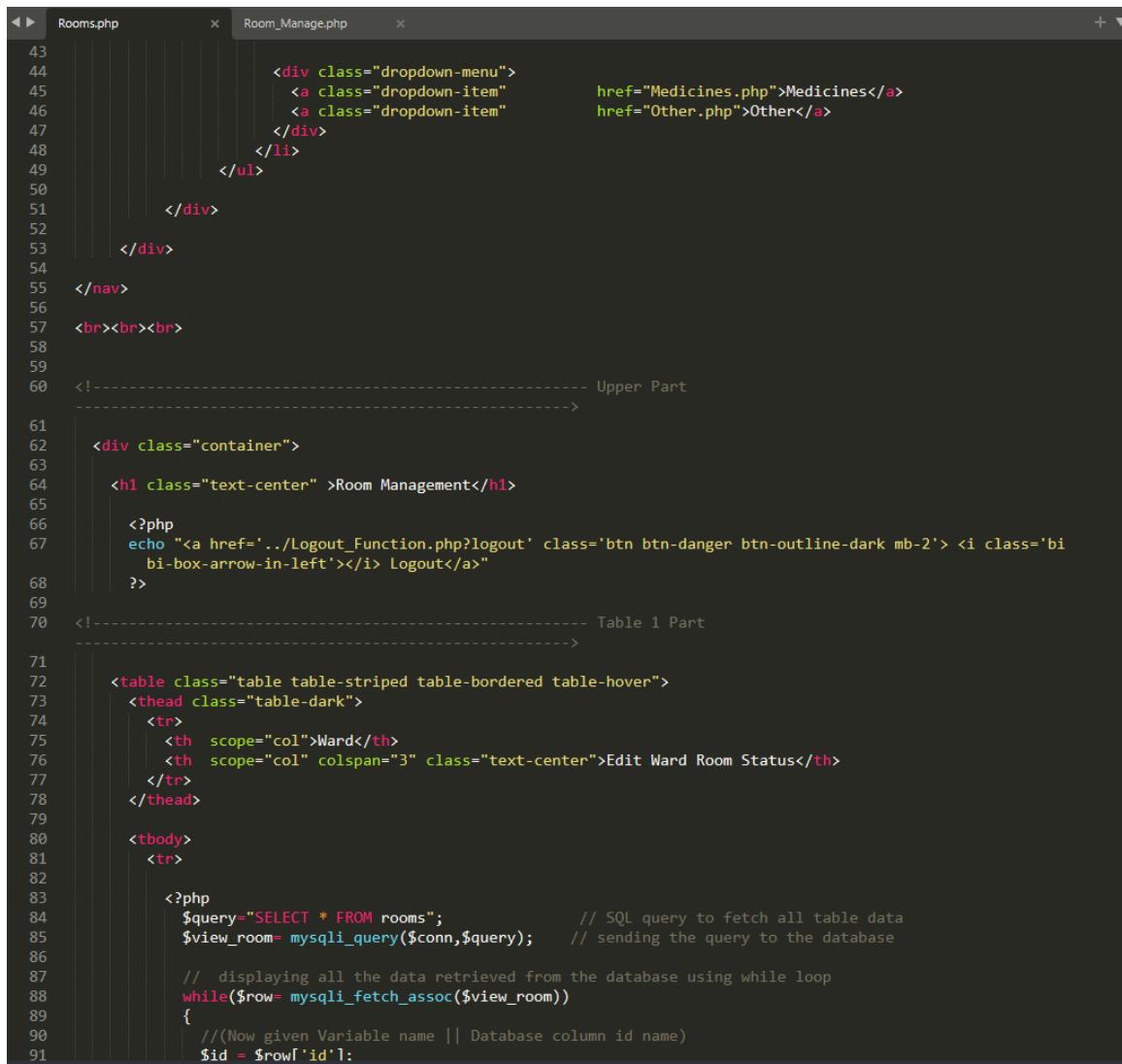
The screenshot shows a web browser window with the URL `localhost/Arogya/Page_Files/Rooms.php`. The page has a teal header bar with the logo 'AROGYA' and navigation links: Home, About Us, Blog, Contact Us, Services, and Supplies. Below the header is a section titled 'Room Management' with a red 'Logout' button. A table titled 'Edit Ward Room Status' lists four wards (Ward 1, Ward 2, Ward 3, Ward 4) each with a 'Manage the Ward Rooms' button. Below this is a section titled 'Ward Rooms Details' with a table showing room availability across four rooms for each ward. A yellow 'Back' button is at the bottom left.

| Ward Number | Room 1 | Room 2 | Room 3 | Room 4 |
|-------------|-----------|-----------|-----------|-----------|
| Ward 1 | No | No | Available | Available |
| Ward 2 | Available | Available | Available | Available |
| Ward 3 | No | No | No | Available |
| Ward 4 | No | No | No | No |

Figure 3. 123 Final Design of Rooms.php

```
1 <?php
2
3     session_start();
4     if(!isset($_SESSION['User'])){
5         header("location:../INDEX.php");
6     }
7
8 ?>
9
10 <!-- Header -->
11 <?php include "../header.php"?>
12
13 <!-- ----- Nav Bar
14 <nav class="navbar navbar-light bg-info navbar-expand-sm fixed-top">
15
16 <div class="container">
17
18     <a href="#" class="navbar-brand" > AROGYA </a>
19
20     <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavDropdown" > <
21     span class="navbar-toggler-icon"></span> </button>
22
23     <div class="collapse navbar-collapse" id="navbarNavDropdown" > <! -- Nav Bar can collapse
24 like a drop down button -->
25
26         <ul class="navbar-nav ms-auto">
27             <li class="nav-item"> <a href="Details_Pages/Home.php" class="nav-link" > Home </a>
28             </li>
29             <li class="nav-item"> <a href="Details_Pages/About.php" class="nav-link" > About Us </a>
30             </li>
31             <li class="nav-item"> <a href="Details_Pages/Blog.php" class="nav-link" > Blog </a>
32             </li>
33             <li class="nav-item"> <a href="Details_Pages/Contact.php" class="nav-link" > Contact Us </a>
34             </li>
35
36             <li class="nav-item dropdown">
37                 <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-toggle="dropdown" > Services </a>
38
39                 <div class="dropdown-menu">
40                     <a class="dropdown-item" href="Staff.php" >Staff</a>
41                     <a class="dropdown-item" href="Patients.php" >Patients</a>
42                     <a class="dropdown-item" href="Wards.php" >Wards</a>
43                     <a class="dropdown-item active" href="Rooms.php" >Rooms</a>
44                 </div>
45             </li>
46
47             <li class="nav-item dropdown">
48                 <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-toggle="dropdown" > Supplies </a>
49             </li>
50         </ul>
51     </div>
52 </div>
53 </nav>
```

Figure 3. 124 Coding for Rooms.php part 1



```

43             <div class="dropdown-menu">
44                 <a class="dropdown-item" href="Medicines.php">Medicines</a>
45                 <a class="dropdown-item" href="Other.php">Other</a>
46             </div>
47         </li>
48     </ul>
49   </div>
50
51   </div>
52
53 <br><br><br>
54
55
56
57 <br><br><br>
58
59
60 <!-- Upper Part
61
62 <div class="container">
63
64     <h1 class="text-center" >Room Management</h1>
65
66     <?php
67     echo "<a href='../Logout_Function.php?logout' class='btn btn-danger btn-outline-dark mb-2'> <i class='bi bi-box-arrow-in-left'></i> Logout</a>" ;
68     ?>
69
70 <!-- Table 1 Part
71
72 <table class="table table-striped table-bordered table-hover">
73     <thead class="table-dark">
74         <tr>
75             <th scope="col">Ward</th>
76             <th scope="col" colspan="3" class="text-center">Edit Ward Room Status</th>
77         </tr>
78     </thead>
79
80     <tbody>
81         <tr>
82
83             <?php
84             $query="SELECT * FROM rooms";           // SQL query to fetch all table data
85             $view_room= mysqli_query($conn,$query);    // sending the query to the database
86
87             // displaying all the data retrieved from the database using while loop
88             while($row= mysqli_fetch_assoc($view_room))
89             {
90                 //((Now given Variable name || Database column id name)
91                 $id = $row['id'];

```

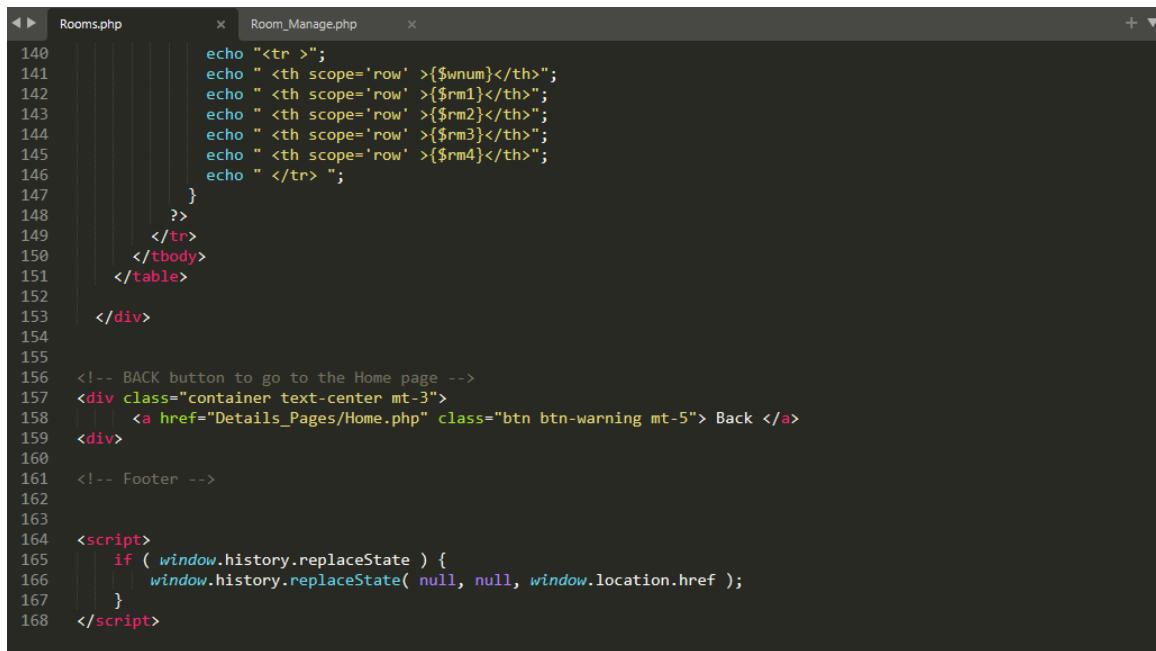
Figure 3. 125 Coding for Rooms.php part 2

```

91     $id = $row['id'];
92     $wnum = $row['ward_id'];
93
94     echo "<tr >";
95     echo " <th scope='row' >{$wnum}</th>";
96
97     echo " <td class='text-center' > <a href='Functions_Room/Room_Manage.php?edit&user_id={$id}' class='btn btn-secondary'><i class='bi bi-building'></i> Manage the Ward Rooms</a> </td> ";
98
99
100    echo " </tr > ";
101   }
102   ?>
103 </tr>
104 </tbody>
105 </table>
106
107 <!-- ----- Table 2 Part
----->
108
109 <h3 class="text-center" >Ward Rooms Details</h3>
110
111 <table class="table table-striped table-bordered table-hover">
112   <thead class="table-dark">
113     <tr>
114       <th scope="col" class="text-center">Ward Number</th>
115       <th scope="col" class="text-center">Room 1</th>
116       <th scope="col" class="text-center">Room 2</th>
117       <th scope="col" class="text-center">Room 3</th>
118       <th scope="col" class="text-center">Room 4</th>
119     </tr>
120   </thead>
121
122   <tbody>
123     <tr>
124
125       <?php
126         $query="SELECT * FROM rooms";           // SQL query to fetch all table data
127         $view_ward= mysqli_query($conn,$query); // sending the query to the database
128
129         // displaying all the data retrieved from the database using while loop
130         while($row= mysqli_fetch_assoc($view_ward))
131         {
132           //((Now given Variable name || Database column id name)
133           $id = $row['id'];
134           $wnum = $row['ward_id'];
135           $rm1 = $row['room1'];
136           $rm2 = $row['room2'];
137           $rm3 = $row['room3'];
138           $rm4 = $row['room4'];
139
140           echo "<tr >":

```

Figure 3. 126 Coding for Rooms.php part 3



The screenshot shows a code editor with two tabs: 'Rooms.php' and 'Room_Manage.php'. The 'Rooms.php' tab is active, displaying the following PHP code:

```
140 echo "<tr >";
141 echo " <th scope='row' >{$wnum}</th>";
142 echo " <th scope='row' >{$rm1}</th>";
143 echo " <th scope='row' >{$rm2}</th>";
144 echo " <th scope='row' >{$rm3}</th>";
145 echo " <th scope='row' >{$rm4}</th>";
146 echo " </tr> ";
147 }
148 ?>
149 </tr>
150 </tbody>
151 </table>
152 </div>
153
154
155
156 <!-- BACK button to go to the Home page -->
157 <div class="container text-center mt-3">
158   <a href="Details_Pages/Home.php" class="btn btn-warning mt-5"> Back </a>
159 </div>
160
161 <!-- Footer -->
162
163
164 <script>
165   if ( window.history.replaceState ) {
166     window.history.replaceState( null, null, window.location.href );
167   }
168 </script>
```

Figure 3. 127 Coding for Rooms.php part 4

Room_Manage.php :

The screenshot shows a web browser window with the URL `localhost/Arogya/Page_Files/Functions_Room/Room_Manage.php...`. The page title is "Ward 1". It displays four room status boxes arranged in a 2x2 grid:

| Room 1 | Room 2 |
|--------|--------|
| No | No |

| Room 3 | Room 4 |
|-----------|-----------|
| Available | Available |

Below the tables are two buttons: a teal "Edit" button and a yellow "Back" button.

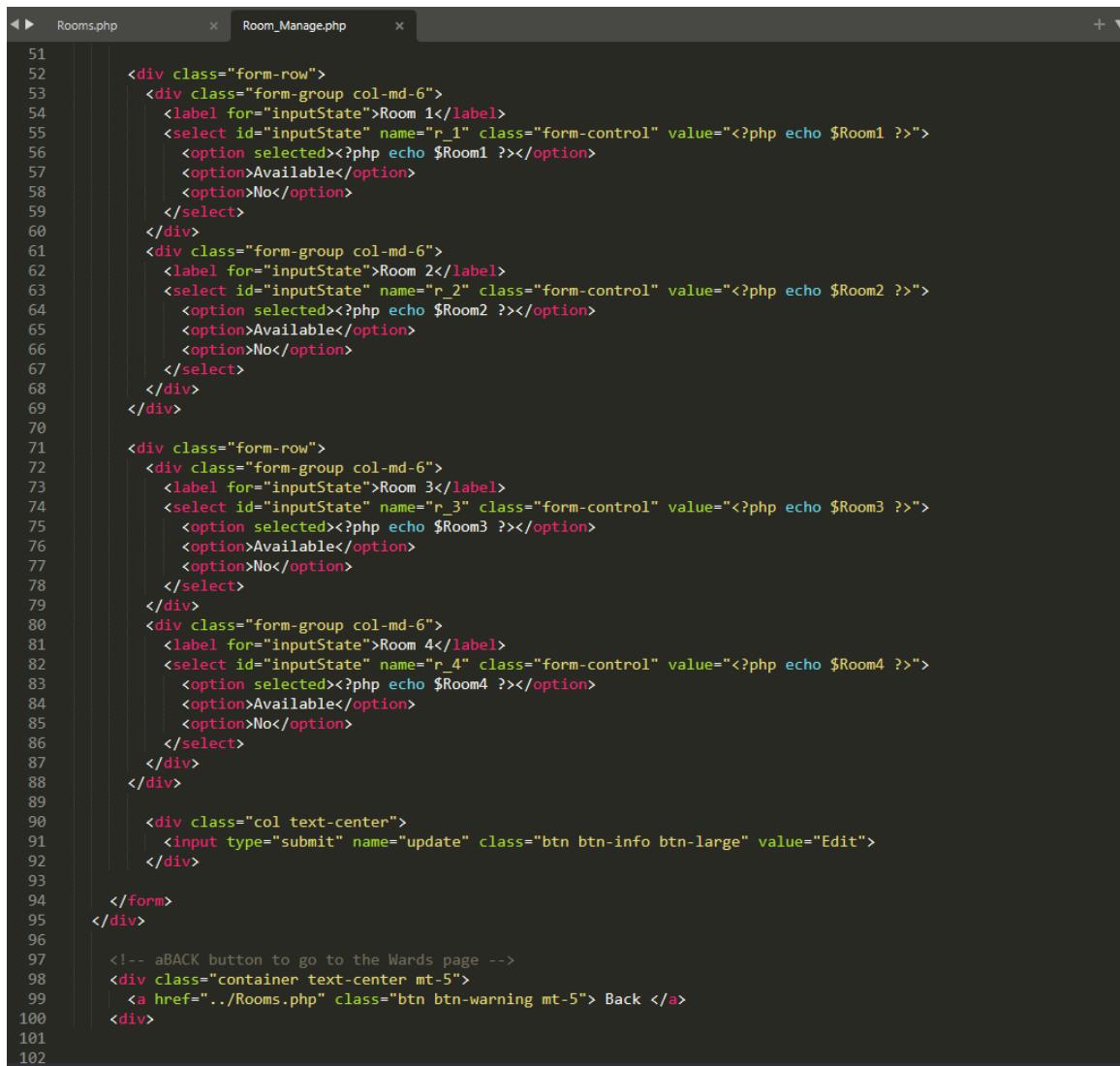
Figure 3. 128 Final Design of Room_Manage.php

```

1 <!-- Footer -->
2 <?php include "../header.php"?>
3
4 <?php
5 // checking if the variable is set or not and if set adding the set data value to variable userid
6 if(isset($_GET['user_id']))
7 {
8     $userid = $_GET['user_id'];
9 }
10 // SQL query to select all the data from the table where id = $userid
11 $query="SELECT * FROM rooms WHERE id = $userid ";
12 $room_manage= mysqli_query($conn,$query);
13
14 while($row = mysqli_fetch_assoc($room_manage))
15 {
16     //((Textbox echo id name || Database column id name)
17     $id = $row['id'];
18     $Wardnum = $row['ward_id'];
19     $Room1 = $row['room1'];
20     $Room2 = $row['room2'];
21     $Room3 = $row['room3'];
22     $Room4 = $row['room4'];
23
24 }
25
26 //Processing form data when form is submitted
27 if(isset($_POST['update']))
28 {
29     //((Now given Variable name || Textbox id name)
30     $RM1 = $_POST['r_1'];
31     $RM2 = $_POST['r_2'];
32     $RM3 = $_POST['r_3'];
33     $RM4 = $_POST['r_4'];
34
35
36     // SQL query to update the data in user table where the id = $userid
37     //((Database column id name || Previously given Variable name)
38     $query = "UPDATE rooms SET room1 = '{$RM1}' , room2 = '{$RM2}' , room3 = '{$RM3}' , room4 = '{$RM4}' WHERE
39         id = $userid";
40     $update_user = mysqli_query($conn, $query);
41     echo "<script type='text/javascript'>alert('Ward room data updated successfully!')</script>";
42     header("location:../Rooms.php");
43 }
44
45
46
47 <h1 class="text-center"><?php echo $Wardnum ?></h1>
48
49 <div class="container " >
50     <form action="" method="post">

```

Figure 3. 129 Coding for Room_Manage.php part 1



```

51      <div class="form-row">
52          <div class="form-group col-md-6">
53              <label for="inputState">Room 1</label>
54              <select id="inputState" name="r_1" class="form-control" value=<?php echo $Room1 ?>">
55                  <option selected><?php echo $Room1 ?></option>
56                  <option>Available</option>
57                  <option>No</option>
58              </select>
59          </div>
60          <div class="form-group col-md-6">
61              <label for="inputState">Room 2</label>
62              <select id="inputState" name="r_2" class="form-control" value=<?php echo $Room2 ?>">
63                  <option selected><?php echo $Room2 ?></option>
64                  <option>Available</option>
65                  <option>No</option>
66              </select>
67          </div>
68      </div>
69
70      <div class="form-row">
71          <div class="form-group col-md-6">
72              <label for="inputState">Room 3</label>
73              <select id="inputState" name="r_3" class="form-control" value=<?php echo $Room3 ?>">
74                  <option selected><?php echo $Room3 ?></option>
75                  <option>Available</option>
76                  <option>No</option>
77              </select>
78          </div>
79          <div class="form-group col-md-6">
80              <label for="inputState">Room 4</label>
81              <select id="inputState" name="r_4" class="form-control" value=<?php echo $Room4 ?>">
82                  <option selected><?php echo $Room4 ?></option>
83                  <option>Available</option>
84                  <option>No</option>
85              </select>
86          </div>
87      </div>
88
89      <div class="col text-center">
90          <input type="submit" name="update" class="btn btn-info btn-large" value="Edit">
91      </div>
92
93  </form>
94</div>
95
96  <!-- aBACK button to go to the Wards page -->
97  <div class="container text-center mt-5">
98      <a href="../Rooms.php" class="btn btn-warning mt-5"> Back </a>
99  </div>
100
101
102

```

Figure 3. 130 Coding for Room_Manage.php part 2

Medicine Stock Management Page

Medicines.php :

The screenshot shows a web browser window with the URL `localhost/Arogya/Page_Files/Medicines.php`. The page title is "AROGYA". The main content area has a heading "Medicine Stock Management". Below it is a table with four rows of data. Each row contains the ID number, name, type, and quantity of a medicine, along with three buttons for "View", "Edit", and "Delete". A "Logout" button is also visible above the table. At the bottom of the page is a "Back" button.

| ID Number | Name | Type | Quantity | Options | | |
|-----------|--------------|--------------|----------|---------|------|--------|
| 1 | Panadol | Antibiotics | 100 | View | Edit | Delete |
| 2 | Paracetamo | Pain Killers | 500 | View | Edit | Delete |
| 3 | Cough Syrup | Other | 50 | View | Edit | Delete |
| 4 | Gastic Syrup | Other | 50 | View | Edit | Delete |

[Back](#)

Figure 3. 131 Final Design of Medicines.php

```

1 <?php
2
3 session_start();
4 if(!isset($_SESSION['User'])){
5 header("location:../INDEX.php");
6 }
7
8 ?>
9
10 <!-- Header -->
11 <?php include "../header.php"?>
12
13 <!-- ----- Nav Bar
14 <nav class="navbar navbar-light bg-info navbar-expand-sm fixed-top">
15
16 <div class="container">
17
18 <a href="#" class="navbar-brand" > AROGYA </a>
19
20 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavDropdown" > <span class="navbar-toggler-icon"></span> </button>
21
22 <div class="collapse navbar-collapse" id="navbarNavDropdown" > ! -- Nav Bar can collapse
like a drop down button -->
23
24 <ul class="navbar-nav ms-auto">
25 <li class="nav-item" > <a href="Details_Pages/Home.php" class="nav-link" > Home </a>
26 <li class="nav-item" > <a href="Details_Pages/About.php" class="nav-link" > About Us </a>
27 <li class="nav-item" > <a href="Details_Pages/Blog.php" class="nav-link" > Blog </a>
28 <li class="nav-item" > <a href="Details_Pages/Contact.php" class="nav-link" > Contact Us </a>
29
30 <li class="nav-item dropdown" >
31 <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-toggle="dropdown" > Services </a>
32
33 <div class="dropdown-menu" >
34 <a class="dropdown-item" href="Staff.php" > Staff </a>
35 <a class="dropdown-item" href="Patients.php" > Patients </a>
36 <a class="dropdown-item" href="Wards.php" > Wards </a>
37 <a class="dropdown-item" href="Rooms.php" > Rooms </a>
38 </div>
39 </li>
40
41 <li class="nav-item dropdown" >
42 <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-toggle="dropdown" > Supplies </a>
43

```

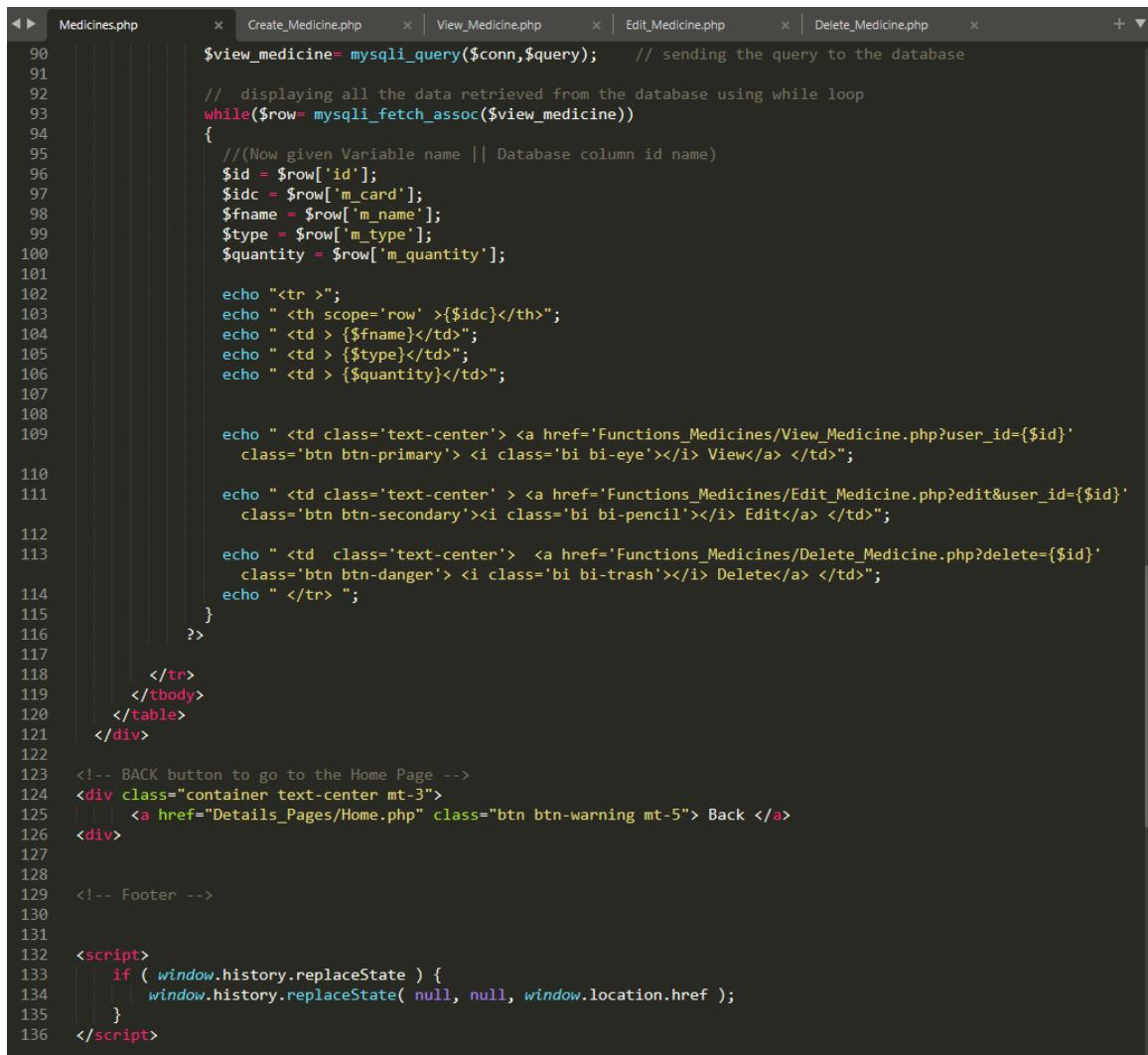
Figure 3. 132 Coding for Medicines.php part 1

```

43             <div class="dropdown-menu">
44                 <a class="dropdown-item active" href="Medicines.php">Medicines</a>
45                 <a class="dropdown-item" href="Other.php">Other</a>
46             </div>
47         </li>
48     </ul>
49   </div>
50
51   </div>
52
53 <br><br><br>
54
55
56
57 <br><br><br>
58
59
60 <!-- Upper Part
61
62 <div class="container">
63
64     <h1 class="text-center" >Medicine Stock Management</h1>
65
66     <a href="Functions_Medicines/Create_Medicine.php" class='btn btn-outline-dark mb-2'> <i class="bi bi-eyedropper"></i> Add New Medicine</a>
67
68     <?php
69     echo "<a href='../Logout_Function.php?logout' class='btn btn-danger btn-outline-dark mb-2'> <i class='bi bi-box-arrow-in-left'></i> Logout</a>">
70     ?>
71
72 <!-- Table Part
73
74 <table class="table table-striped table-bordered table-hover">
75     <thead class="table-dark">
76         <tr>
77             <th scope="col" class="text-center">ID Number</th>
78             <th scope="col" class="text-center">Name</th>
79             <th scope="col" class="text-center">Type</th>
80             <th scope="col" class="text-center">Quantity</th>
81             <th scope="col" colspan="3" class="text-center">Options</th>
82         </tr>
83     </thead>
84
85     <tbody>
86         <tr>
87
88             <?php
89                 $query="SELECT * FROM medicine";           // SQL query to fetch all table data
90                 $view_medicine= mysqli_query($conn,$query); // sending the query to the database

```

Figure 3. 133 Coding for Medicines.php part 2



```

90     $view_medicine= mysqli_query($conn,$query); // sending the query to the database
91
92     // displaying all the data retrieved from the database using while loop
93     while($row= mysqli_fetch_assoc($view_medicine))
94     {
95         //Now given Variable name || Database column id name
96         $id = $row['id'];
97         $idc = $row['m_card'];
98         $fname = $row['m_name'];
99         $type = $row['m_type'];
100        $quantity = $row['m_quantity'];
101
102        echo "<tr >";
103        echo " <th scope='row' >{$idc}</th>";
104        echo " <td > {$fname}</td>";
105        echo " <td > {$type}</td>";
106        echo " <td > {$quantity}</td>";
107
108        echo " <td class='text-center'> <a href='Functions_Medicines/View_Medicine.php?user_id={$id}'"
109             class='btn btn-primary'> 

```

Figure 3. 134 Coding for Medicines.php part 3

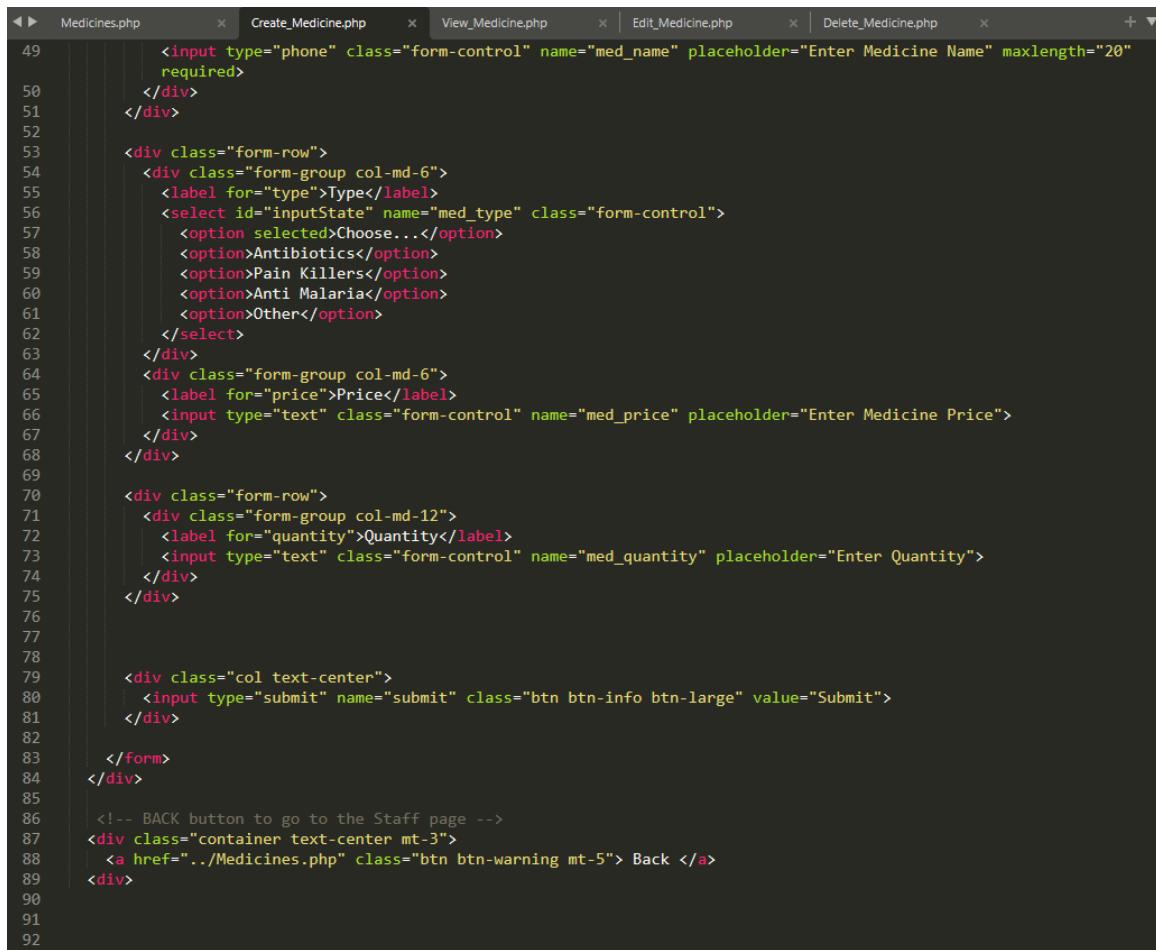
Create_Medicine.php :

The screenshot shows a web browser window with the URL `localhost/Arogya/Page_Files/Functions_Medicines/Create_Medicine.php`. The page title is "Add Medicine Details". The form contains four input fields: "ID Number" (with placeholder "Enter ID Number"), "Medicine Name" (with placeholder "Enter Medicine Name"), "Type" (with placeholder "Choose..."), and "Price" (with placeholder "Enter Medicine Price"). Below these is a "Quantity" field with placeholder "Enter Quantity". At the bottom are a "Submit" button (blue) and a "Back" button (yellow).

Figure 3. 135 Final Design of Create_Medicine.php

```
1 <!-- Header -->
2 <?php include "../../header.php" ?>
3
4 <?php
5
6 //To give new values to DATABASE
7 if(isset($_POST['submit']))
8 {
9     //Now given Variable name || Textbox id name)
10    $id = $_POST['med_card'];
11    $Name = $_POST['med_name'];
12    $Type = $_POST['med_type'];
13    $Price = $_POST['med_price'];
14    $Quantity = $_POST['med_quantity'];
15
16
17
18 // SQL query to insert user data into the users table
19 //Database column id name || Previously given Variable name)
20 $query= "INSERT INTO medicine(m_card, m_name, m_type, m_price, m_quantity) VALUES('{$id}', '{$Name}', '{$Type}', '{$Price}', '{$Quantity}')";
21 $add_medicine = mysqli_query($conn,$query);
22
23 // displaying proper message for the user to see whether the query executed perfectly or not
24 if ($add_medicine)
25 {
26     echo "something went wrong ". mysqli_error($conn);
27 }
28
29 else
30 { echo "<script type='text/javascript'>alert('Medicine added successfully!')</script>";
31     header('Location:../../Medicines.php');
32 }
33
34 }
35 ?>
36
37
38 <h1 class="text-center">Add Medicine Details</h1>
39 <div class="container">
40     <form action="" method="post">
41
42         <div class="form-row">
43             <div class="form-group col-md-6">
44                 <label for="idnumber">ID Number</label>
45                 <input type="text" class="form-control" name="med_card" placeholder="Enter ID Number" maxlength="12" required>
46             </div>
47             <div class="form-group col-md-6">
48                 <label for="name">Medicine Name</label>
49                 <input type="text" class="form-control" name="med_name" placeholder="Enter Medicine Name" maxlength="20" required>
50             </div>
51         </div>
52         <div class="form-group">
53             <label for="price">Price</label>
54             <input type="text" class="form-control" name="med_price" placeholder="Enter Price" required>
55         </div>
56         <div class="form-group">
57             <label for="quantity">Quantity</label>
58             <input type="text" class="form-control" name="med_quantity" placeholder="Enter Quantity" required>
59         </div>
60
61         <div class="form-group">
62             <label for="type">Type</label>
63             <input type="text" class="form-control" name="med_type" placeholder="Enter Type" required>
64         </div>
65
66         <div class="form-group">
67             <label for="submit">Submit</label>
68             <input type="submit" class="form-control" value="Submit" name="submit" required>
69         </div>
70     </form>
71 </div>
```

Figure 3. 136 Coding for Create_Medicine.php part 1



```
49     <input type="phone" class="form-control" name="med_name" placeholder="Enter Medicine Name" maxlength="20"
50     required>
51 </div>
52 </div>
53 <div class="form-row">
54     <div class="form-group col-md-6">
55         <label for="type">Type</label>
56         <select id="inputState" name="med_type" class="form-control">
57             <option selected>Choose...</option>
58             <option>Antibiotics</option>
59             <option>Pain Killers</option>
60             <option>Anti Malaria</option>
61             <option>Other</option>
62         </select>
63     </div>
64     <div class="form-group col-md-6">
65         <label for="price">Price</label>
66         <input type="text" class="form-control" name="med_price" placeholder="Enter Medicine Price">
67     </div>
68 </div>
69 <div class="form-row">
70     <div class="form-group col-md-12">
71         <label for="quantity">Quantity</label>
72         <input type="text" class="form-control" name="med_quantity" placeholder="Enter Quantity">
73     </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 <div class="col text-center">
79     <input type="submit" name="submit" class="btn btn-info btn-large" value="Submit">
80 </div>
81 </div>
82 </form>
83 </div>
84 </div>
85 <!-- BACK button to go to the Staff page -->
86 <div class="container text-center mt-3">
87     <a href="../Medicines.php" class="btn btn-warning mt-5"> Back </a>
88 <div>
89 <div>
90 <div>
91 <div>
92 <div>
```

Figure 3. 137 Coding for Create_Medicine.php part 2

View_Medicine.php :

```

1  <!-- Header -->
2  <?php include '../header.php'?>
3
4  <h1 class="text-center">Staff Details</h1>
5
6  <div class="container">
7
8      <table class="table table-striped table-bordered table-hover">
9          <thead class="table-dark">
10             <tr>
11                 <th scope="col">ID</th>
12                 <th scope="col">Medicine Name</th>
13                 <th scope="col">Type</th>
14                 <th scope="col">Price</th>
15                 <th scope="col">Quantity</th>
16             </tr>
17         </thead>
18         <tbody>
19             <tr>
20                 <?php
21                     // first we check using 'isset()' function if the variable is set or not
22                     //Processing form data when form is submitted
23                     if (isset($_GET['user_id']))
24                 {
25                     $userid = $_GET['user_id'];
26
27                     // SQL query to fetch the data where id=$userid & storing data in view_user
28                     $query="SELECT * FROM medicine WHERE id = {$userid} ";
29                     $view_medicine= mysqli_query($conn,$query);
30
31                     while($row = mysqli_fetch_assoc($view_medicine))
32                     {
33                         //Now given Variable name || Database column id name
34                         $id = $row['id'];
35                         $idc = $row['m_card'];
36                         $Name = $row['m_name'];
37                         $Type = $row['m_type'];
38                         $Price = $row['m_price'];
39                         $Quantity = $row['m_quantity'];
40
41                         echo "<tr >";
42                         echo " <td >{$idc}</td>";
43                         echo " <td > {$Name}</td>";
44                         echo " <td > {$Type}</td>";
45                         echo " <td > {$Price} </td>";
46                         echo " <td > {$Quantity}</td>";
47                         echo " </tr> ";
48                     }
49                 }
50             </tr>
51         </tbody>
52     </table>

```

Figure 3. 138 Coding for View_Medicine.php part 1

```
52 |     |     | </tbody>
53 |     |     | </table>
54 |
55 |     | </div>
56 |
57 |
58 |     | <!-- BACK Button to go to previous page -->
59 |     | <div class="container text-center mt-5">
60 |     |     | <a href="../Medicines.php" class="btn btn-warning mt-5"> Back </a>
61 |     | </div>
62 |
63 |     | <!-- Footer -->
64 |     | <?php include "../../footer.php" ?>
```

Figure 3. 139 Coding for View_Medicine.php part 2

Edit_Medicine.php :

The screenshot shows a web browser window with the title 'Edit Medicine Details'. The page contains four input fields: 'ID Number' (value 1), 'Medicine Name' (value Panadol), 'Type' (value Antibiotics), and 'Price' (value 5). Below these is a 'Quantity' field containing the value 100. At the bottom are two buttons: a teal 'Edit' button and a yellow 'Back' button.

| | |
|-------------|---------------|
| ID Number | Medicine Name |
| 1 | Panadol |
| Type | Price |
| Antibiotics | 5 |
| Quantity | |
| 100 | |

Edit **Back**

Figure 3. 140 Final Design of Edit_Medicine.php

```

1 <!-- Header -->
2 <?php include "../../header.php"?>
3
4 <?php
5 // checking if the variable is set or not and if set adding the set data value to variable userid
6 if(isset($_GET['user_id']))
7 {
8     $userid = $_GET['user_id'];
9 }
10 // SQL query to select all the data from the table where id = $userid ()
11 $query="SELECT * FROM medicine WHERE id = $userid ";
12 $edit_medicine= mysqli_query($conn,$query);
13
14 //To see DATABASE values
15 while($row = mysqli_fetch_assoc($edit_medicine))
16 {
17     //Textbox echo id name || Database column id name
18     $id = $row['id'];
19     $mdc = $row['m_card'];
20     $mName = $row['m_name'];
21     $mType = $row['m_type'];
22     $mPrice = $row['m_price'];
23     $mQuantity = $row['m_quantity'];
24 }
25
26 //To give new values to DATABASE
27 if(isset($_POST['update']))
28 {
29     //Now given Variable name || Textbox id name
30     $ID = $_POST['med_no'];
31     $Name = $_POST['med_name'];
32     $Type = $_POST['med_type'];
33     $Price = $_POST['med_price'];
34     $Quantity = $_POST['med_quantity'];
35
36     // SQL query to update the data in user table where the id = $userid
37     //Database column id name || Previously given Variable name
38     $query = "UPDATE medicine SET m_card = '{$ID}' , m_name = '{$Name}' , m_type = '{$Type}' , m_price = '{$Price}'
39     , m_quantity = '{$Quantity}' WHERE id = $userid";
40     $update_user = mysqli_query($conn, $query);
41     echo "<script type='text/javascript'>alert('Medicine edited successfully!')</script>";
42     header("location:../Medicines.php");
43 }
44
45 <h1 class="text-center">Edit Medicine Details</h1>
46 <div class="container " >
47 <form action="" method="post">
48
49     <div class="form-row">
50         <div class="form-group col-md-6">
51             <label for="id">ID Number</label>

```

Figure 3. 141 Coding for Edit_Medicine.php part 1

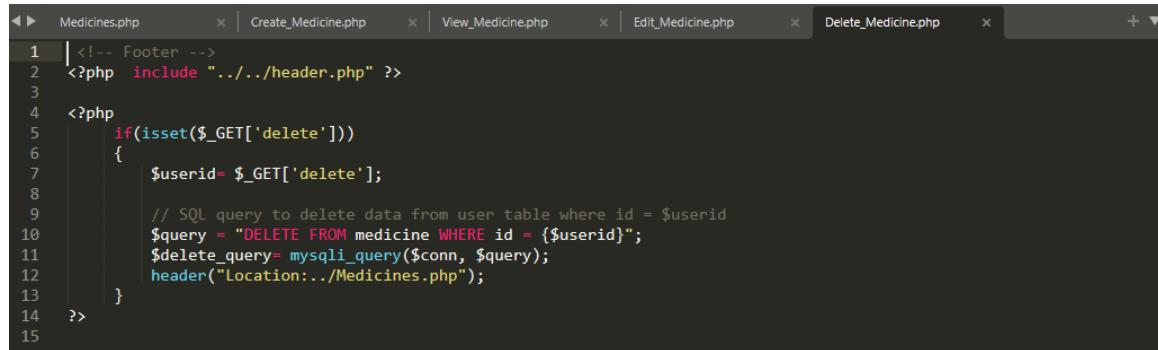
```

51     <label for="id">ID Number</label>
52     <input type="text" class="form-control" name="med_no" value=<?php echo $mdc ?>" maxlength="12" required>
53   </div>
54   <div class="form-group col-md-6">
55     <label for="name">Medicine Name</label>
56     <input type="phone" class="form-control" name="med_name" value=<?php echo $mName ?>" maxlength="10" required>
57   </div>
58 </div>
59
60 <div class="form-row">
61   <div class="form-group col-md-6">
62     <label for="type">Type</label>
63     <select id="inputState" name="med_type" class="form-control" value=<?php echo $mType ?>">
64       <option selected><?php echo $mType ?></option>
65       <option>Antibiotics</option>
66       <option>Pain Killers</option>
67       <option>Anti Malaria</option>
68       <option>Other</option>
69     </select>
70   </div>
71   <div class="form-group col-md-6">
72     <label for="price">Price</label>
73     <input type="text" class="form-control" name="med_price" value=<?php echo $mPrice ?>">
74   </div>
75 </div>
76
77 <div class="form-row">
78   <div class="form-group col-md-12">
79     <label for="quantity">Quantity</label>
80     <input type="text" class="form-control" name="med_quantity" value=<?php echo $mQuantity ?>">
81   </div>
82 </div>
83
84
85
86   <div class="col text-center">
87     <input type="submit" name="update" class="btn btn-info btn-large" value="Edit">
88   </div>
89
90 </form>
91 </div>
92
93 <!-- BACK button to go to the Staff page -->
94 <div class="container text-center mt-5">
95   <a href="../Medicines.php" class="btn btn-warning mt-5"> Back </a>
96 </div>
97
98
99
100 <!-- Footer -->
101 <?php include "../../footer.php" ?>

```

Figure 3. 142 Coding for Edit_Medicine.php part 2

Delete_Medicine.php :



```
1 | <!-- Footer -->
2 | <?php include "../header.php" ?>
3 |
4 | <?php
5 |     if(isset($_GET['delete']))
6 |     {
7 |         $userid= $_GET['delete'];
8 |
9 |         // SQL query to delete data from user table where id = $userid
10 |        $query = "DELETE FROM medicine WHERE id = {$userid}";
11 |        $delete_query= mysqli_query($conn, $query);
12 |        header("Location:../Medicines.php");
13 |
14 |    }
15 | ?>
```

Figure 3. 143 Coding for Delete_Medicine.php part 1

Other supplies Management

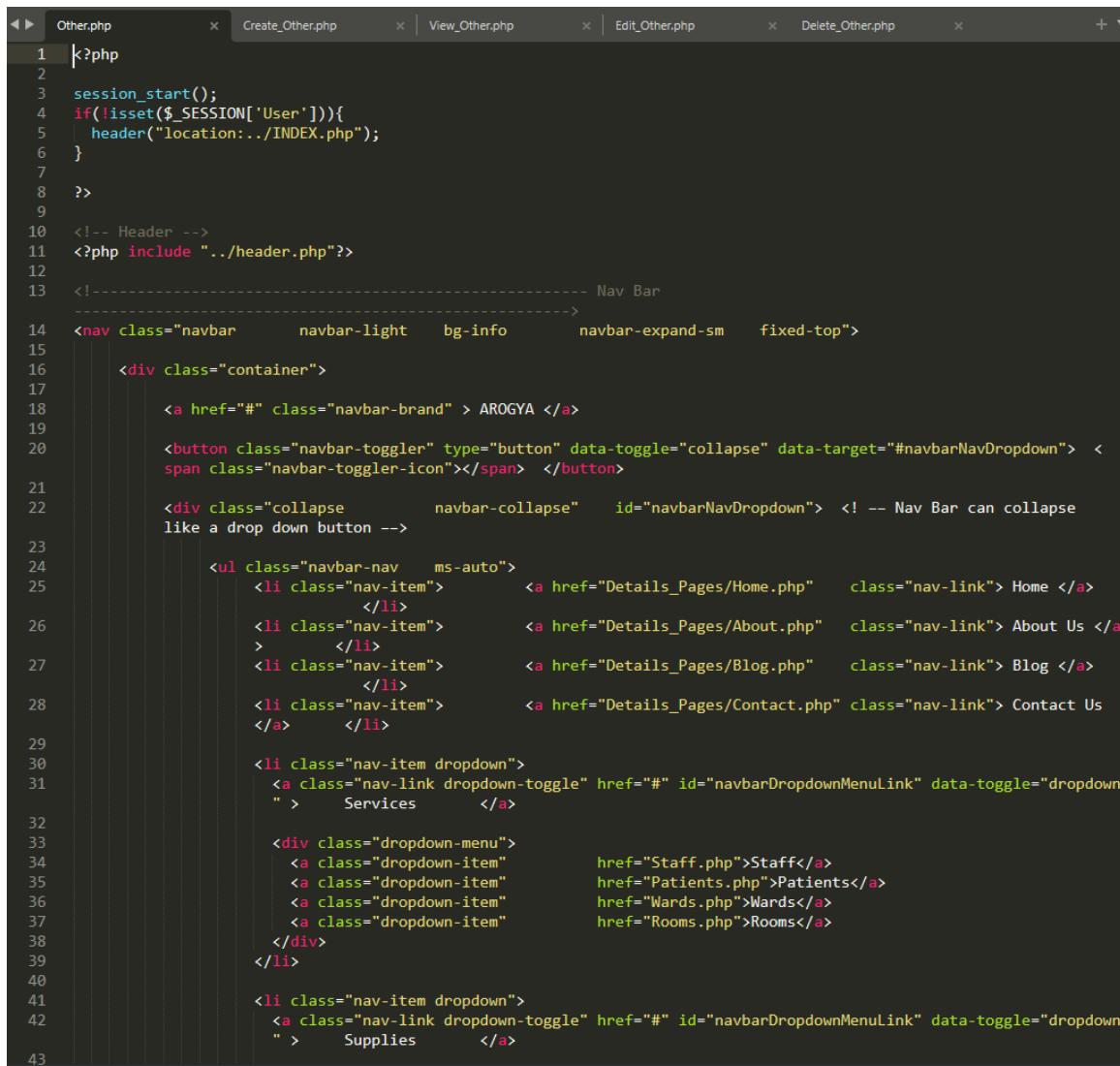
Other.php :

The screenshot shows a web browser window with the URL `localhost/Arogya/Page_Files/Other.php`. The page title is "AROGYA". The main content area has a heading "Other Supplies Management". Below the heading are two buttons: "Add New Item" (with a plus icon) and "Logout". A table displays four items:

| ID Number | Name | Type | Quantity | Options | | |
|-----------|-----------|--------------------|----------|---------|------|--------|
| 1 | Pen | Office Stationary | 20 | View | Edit | Delete |
| 2 | Bed | Beds and Metresses | 5 | View | Edit | Delete |
| 3 | Metresses | Beds and Metresses | 10 | View | Edit | Delete |
| 4 | Van | Vehicles | 2 | View | Edit | Delete |

A yellow "Back" button is located at the bottom right of the content area.

Figure 3. 144 Final Design of Other.php



The screenshot shows a code editor window with multiple tabs at the top: 'Other.php', 'Create_Other.php', 'View_Other.php', 'Edit_Other.php', and 'Delete_Other.php'. The 'Other.php' tab is active. The code is a PHP script with embedded HTML and CSS. It starts with session handling and includes a header file. Below that is a navigation bar (navbar) with a brand link, a toggle button, and a dropdown menu containing links for Home, About Us, Blog, Contact Us, Services (with sub-links for Staff, Patients, Wards, Rooms), and Supplies.

```
<?php
session_start();
if(!isset($_SESSION['User'])){
header("location:../INDEX.php");
}
?>
<!-- Header -->
<?php include "../header.php"?>
<!-- Nav Bar
<nav class="navbar navbar-light bg-info navbar-expand-sm fixed-top">


-


```

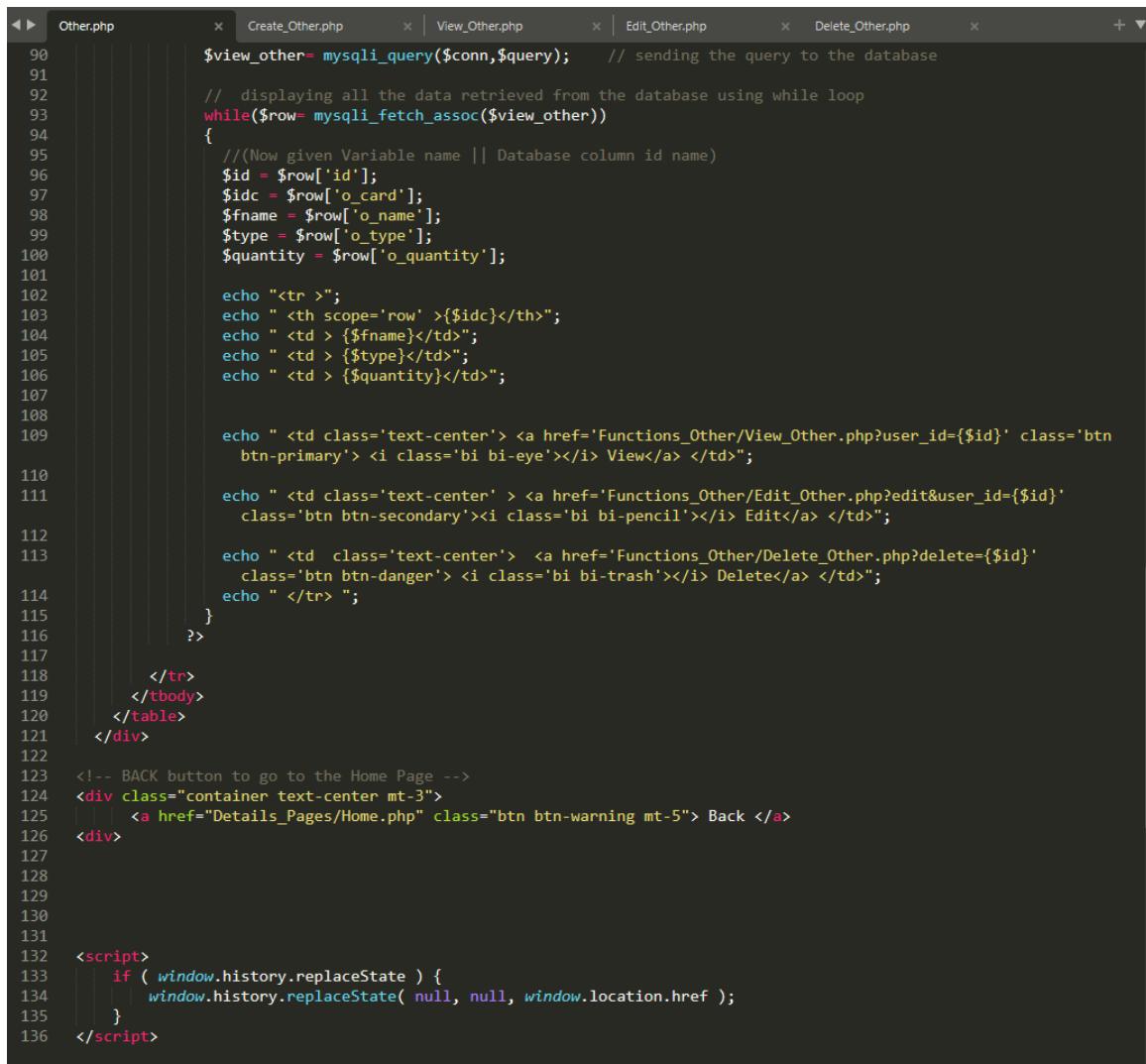
Figure 3. 145 Coding for Other.php part 1

```

43             <div class="dropdown-menu">
44                 <a class="dropdown-item" href="Medicines.php">Medicines</a>
45                 <a class="dropdown-item active" href="Other.php">Other</a>
46             </div>
47         </li>
48     </ul>
49   </div>
50
51   <br><br><br>
52
53 </nav>
54
55 <br><br><br>
56
57 <!-- Upper Part -->
58
59
60 <!-- Table Part -->
61
62 <div class="container">
63
64     <h1 class="text-center" >Other Supplies Management</h1>
65
66     <a href="Functions_Other/Create_Other.php" class='btn btn-outline-dark mb-2'> <i class="bi bi-box-seam"></i>
67     Add New Item</a>
68
69     <?php
70     echo "<a href='../Logout_Function.php?logout' class='btn btn-danger btn-outline-dark mb-2'> <i class='bi bi-box-arrow-in-left'></i> Logout</a>" ;
71     ?>
72 <!-- Table Part -->
73
74 <table class="table table-striped table-bordered table-hover">
75     <thead class="table-dark">
76         <tr>
77             <th scope="col" class="text-center">ID Number</th>
78             <th scope="col" class="text-center">Name</th>
79             <th scope="col" class="text-center">Type</th>
80             <th scope="col" class="text-center">Quantity</th>
81             <th scope="col" colspan="3" class="text-center">Options</th>
82         </tr>
83     </thead>
84
85     <tbody>
86         <tr>
87
88             <?php
89                 $query="SELECT * FROM other";           // SQL query to fetch all table data
90                 $view other= mysqli_query($conn,$query); // sending the query to the database

```

Figure 3. 146 Coding for Other.php part 2



```

90     $view_other= mysqli_query($conn,$query);      // sending the query to the database
91
92     // displaying all the data retrieved from the database using while loop
93     while($row= mysqli_fetch_assoc($view_other))
94     {
95         //Now given Variable name || Database column id name)
96         $id = $row['id'];
97         $idc = $row['o_card'];
98         $fname = $row['o_name'];
99         $type = $row['o_type'];
100        $quantity = $row['o_quantity'];
101
102        echo "<tr >";
103        echo " <th scope='row' >{$idc}</th>";
104        echo " <td > {$fname}</td>";
105        echo " <td > {$type}</td>";
106        echo " <td > {$quantity}</td>";
107
108        echo " <td class='text-center'> <a href='Functions_Other/View_Other.php?user_id={$id}' class='btn
109          btn-primary'> <i class='bi bi-eye'></i> View</a> </td>";
110
111        echo " <td class='text-center' > <a href='Functions_Other/Edit_Other.php?edit&user_id={$id}'>
112          <i class='bi bi-pencil'></i> Edit</a> </td>";
113
114        echo " <td class='text-center'> <a href='Functions_Other/Delete_Other.php?delete={$id}'>
115          <i class='bi bi-trash'></i> Delete</a> </td>";
116        echo " </tr> ";
117
118    </tr>
119    </tbody>
120    </table>
121    </div>
122
123    <!-- BACK button to go to the Home Page --&gt;
124    &lt;div class="container text-center mt-3"&gt;
125        &lt;a href="Details_Pages/Home.php" class="btn btn-warning mt-5"&gt; Back &lt;/a&gt;
126    &lt;/div&gt;
127
128
129
130
131
132    &lt;script&gt;
133        if ( window.history.replaceState ) {
134            window.history.replaceState( null, null, window.location.href );
135        }
136    &lt;/script&gt;
</pre>

```

Figure 3. 147 Coding for Other.php part 3

Create_Other.php :

The screenshot shows a web browser window with the URL `localhost/Arogya/Page_Files/Functions_Other/Create_Other.php`. The page title is "Add Item Details". The form consists of four input fields arranged in a grid:

| ID Number | Item Name |
|-----------------|------------------|
| Enter ID Number | Enter Item Name |
| Item Type | Price |
| Choose... | Enter Item Price |
| Quantity | |
| Enter Quantity | |

Below the form are two buttons: a teal "Submit" button and a yellow "Back" button.

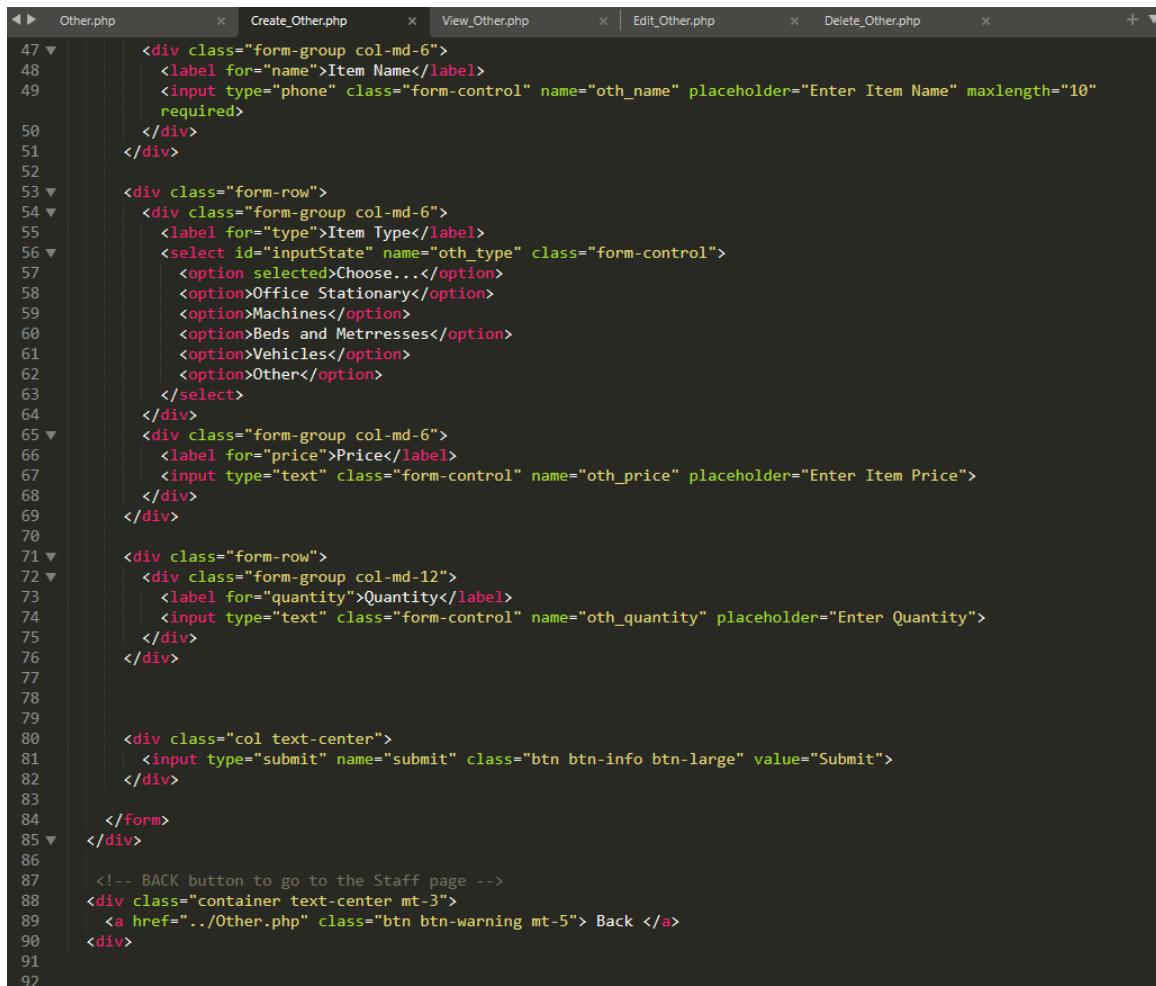
Figure 3. 148 Final Design of Create_Other.php

```

1 |<!-- Header -->
2 |<?php include "../header.php" ?>
3 |
4 |<?php
5 |
6 | //To give new values to DATABASE
7 | if(isset($_POST['submit']))
8 | {
9 |     //Now given Variable name || Textbox id name
10 |     $id = $_POST['oth_card'];
11 |     $Name = $_POST['oth_name'];
12 |     $Type = $_POST['oth_type'];
13 |     $Price = $_POST['oth_price'];
14 |     $Quantity = $_POST['oth_quantity'];
15 |
16 |
17 |
18 |     // SQL query to insert user data into the users table
19 |     // (Database column id name || Previously given Variable name)
20 |     $query= "INSERT INTO other(o_card, o_name, o_type, o_price, o_quantity) VALUES('{$id}', '{$Name}', '{$Type}', '{$Price}', '{$Quantity}')";
21 |     $add_other = mysqli_query($conn,$query);
22 |
23 |     // displaying proper message for the user to see whether the query executed perfectly or not
24 |     if (!$add_other)
25 |     {
26 |         echo "something went wrong ". mysqli_error($conn);
27 |     }
28 |
29 |
30 |     else
31 |     { echo "<script type='text/javascript'>alert('Item added successfully')</script>";
32 |       header('Location:../Other.php');
33 |     }
34 |
35 | }
36 |
37 |
38 | <h1 class="text-center">Add Item Details</h1>
39 | <div class="container">
40 |   <form action="" method="post">
41 |
42 |     <div class="form-row">
43 |       <div class="form-group col-md-6">
44 |         <label for="idnumber">ID Number</label>
45 |         <input type="text" class="form-control" name="oth_card" placeholder="Enter ID Number" maxlength="12"
46 |             required>
47 |       </div>
48 |       <div class="form-group col-md-6">
49 |         <label for="name">Item Name</label>
        <input type="phone" class="form-control" name="oth_name" placeholder="Enter Item Name" maxlength="10"
            required>
      </div>
    </div>
  </form>
</div>

```

Figure 3. 149 Coding for Create_Other.php part I



```
47 ▼    <div class="form-group col-md-6">
48      <label for="name">Item Name</label>
49      <input type="phone" class="form-control" name="oth_name" placeholder="Enter Item Name" maxlength="10"
50          required>
51    </div>
52
53 ▼    <div class="form-row">
54      <div class="form-group col-md-6">
55          <label for="type">Item Type</label>
56          <select id="inputState" name="oth_type" class="form-control">
57              <option selected>Choose...</option>
58              <option>Office Stationary</option>
59              <option>Machines</option>
60              <option>Beds and Metrresses</option>
61              <option>Vehicles</option>
62              <option>Other</option>
63          </select>
64      </div>
65      <div class="form-group col-md-6">
66          <label for="price">Price</label>
67          <input type="text" class="form-control" name="oth_price" placeholder="Enter Item Price">
68      </div>
69  </div>
70
71 ▼    <div class="form-row">
72      <div class="form-group col-md-12">
73          <label for="quantity">Quantity</label>
74          <input type="text" class="form-control" name="oth_quantity" placeholder="Enter Quantity">
75      </div>
76  </div>
77
78
79
80      <div class="col text-center">
81          <input type="submit" name="submit" class="btn btn-info btn-large" value="Submit">
82      </div>
83
84  </form>
85  </div>
86
87  <!-- BACK button to go to the Staff page -->
88  <div class="container text-center mt-3">
89      <a href="../Other.php" class="btn btn-warning mt-5"> Back </a>
90  </div>
91
92
```

Figure 3. 150 Coding for Create_Other.php part 2

View_Other.php :

localhost/Arogya/Page_Files/Functions_Other/View_Other.php?u...

Staff Details

| ID | Item Name | Item Type | Price | Quantity |
|----|-----------|-------------------|-------|----------|
| 1 | Pen | Office Stationary | 10 | 20 |

Back

Figure 3. 151 Final Design of View_Other.php

```

1 |<!-- Header -->
2 |<?php include '../header.php'>
3 |
4 |<h1 class="text-center">Staff Details</h1>
5 |
6 |<div class="container">
7 |
8 |    <table class="table table-striped table-bordered table-hover">
9 |        <thead class="table-dark">
10 |            <tr>
11 |                <th scope="col">ID</th>
12 |                <th scope="col">Item Name</th>
13 |                <th scope="col">Item Type</th>
14 |                <th scope="col">Price</th>
15 |                <th scope="col">Quantity</th>
16 |            </tr>
17 |        </thead>
18 |        <tbody>
19 |            <tr>
20 |                <?php
21 |                    // first we check using 'isset()' function if the variable is set or not
22 |                    //Processing form data when form is submitted
23 |                    if (isset($_GET['user_id'])) {
24 |                        {
25 |                            $userid = $_GET['user_id'];
26 |
27 |                            // SQL query to fetch the data where id=$userid & storing data in view_other
28 |                            $query="SELECT * FROM other WHERE id = {$userid} ";
29 |                            $view_other= mysqli_query($conn,$query);
30 |
31 |                            while($row = mysqli_fetch_assoc($view_other))
32 |                            {
33 |                                //Now given Variable name || Database column id name
34 |                                $id = $row['id'];
35 |                                $idc = $row['o_card'];
36 |                                $Name = $row['o_name'];
37 |                                $Type = $row['o_type'];
38 |                                $Price = $row['o_price'];
39 |                                $Quantity = $row['o_quantity'];
40 |
41 |                                echo "<tr >";
42 |                                echo " <td >{$idc}</td>";
43 |                                echo " <td > {$Name}</td>";
44 |                                echo " <td > {$Type}</td>";
45 |                                echo " <td > {$Price} </td>";
46 |                                echo " <td > {$Quantity}</td>";
47 |                                echo " </tr> ";
48 |                            }
49 |                        }
50 |                    ?>
51 |                </tr>
52 |            </tbody>

```

Figure 3. 152 Coding for View_Other.php part 1

```
49     }
50     ?>
51     <tr>
52     </tbody>
53   </table>
54 
55 </div>
56 
57 <!-- BACK Button to go to previous page -->
58 <div class="container text-center mt-5">
59   <a href="../Other.php" class="btn btn-warning mt-5"> Back </a>
60 </div>
61 
62 <!-- Footer -->
63 <?php include " ../../footer.php" ?>
```

Figure 3. 153 Coding for View_Other.php part 2

Edit_Other.php :

The screenshot shows a web browser window with the URL `localhost/Arogya/Page_Files/Functions_Other/Edit_Other.php?ed...`. The page title is "Edit Item Details". The form contains the following fields:

| ID Number | Item Name |
|-----------|-----------|
| 1 | Pen |

| Type | Price |
|-------------------|-------|
| Office Stationary | 10 |

| Quantity |
|----------|
| 20 |

Edit (button)

Back (button)

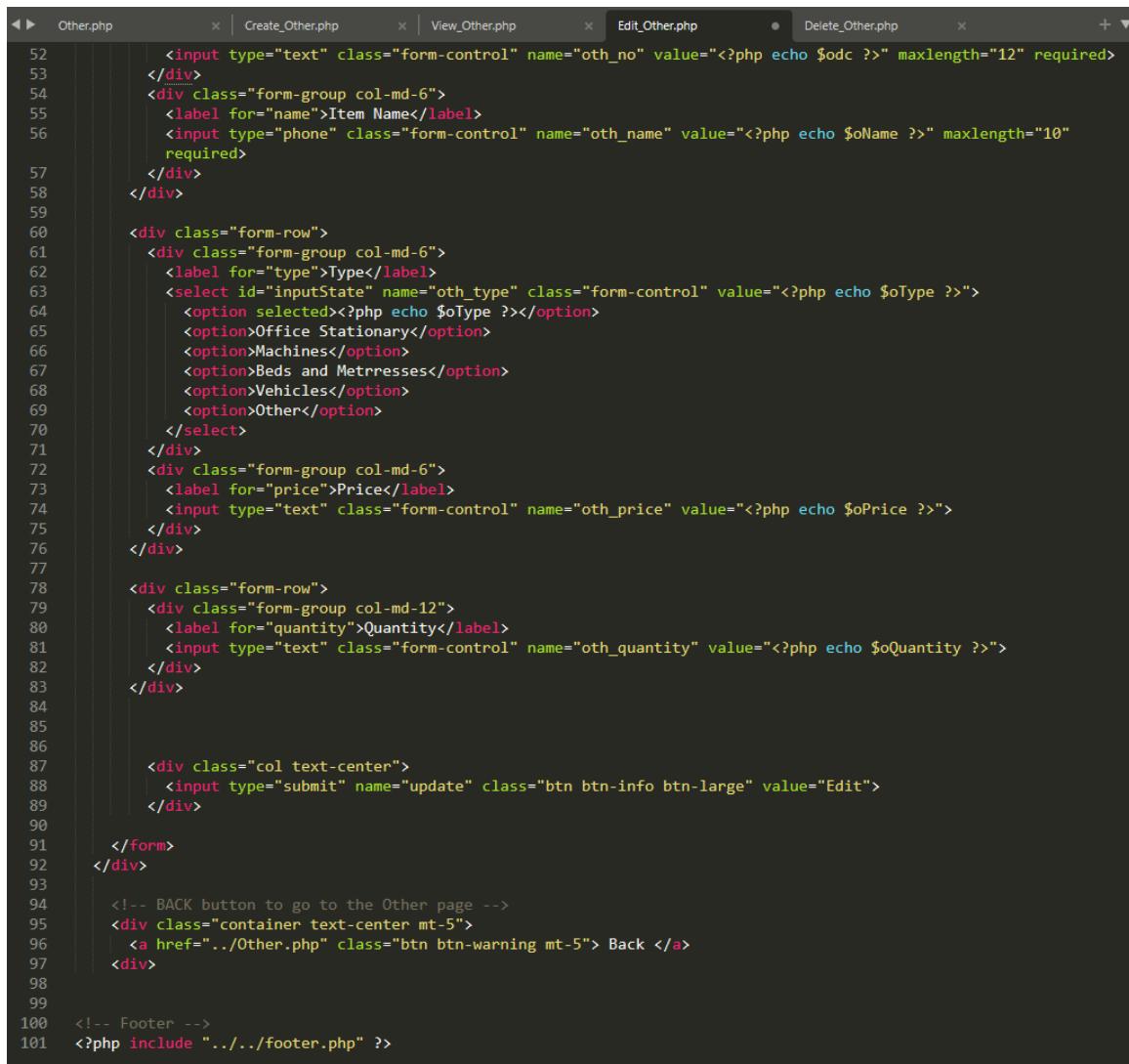
Figure 3. 154 Final Design of Edit_Other.php

```

1 <!-- Header -->
2 <?php include "../header.php"?>
3
4 <?php
5     // checking if the variable is set or not and if set adding the set data value to variable userid
6     if(isset($_GET['user_id']))
7     {
8         $userid = $_GET['user_id'];
9     }
10    // SQL query to select all the data from the table where id = $userid ()
11    $query="SELECT * FROM other WHERE id = $userid ";
12    $edit_other= mysqli_query($conn,$query);
13
14    //To see DATABASE values
15    while($row = mysqli_fetch_assoc($edit_other))
16    {
17        //Textbox echo id name || Database column id name
18        $id = $row['id'];
19        $oCard = $row['o_card'];
20        $oName = $row['o_name'];
21        $oType = $row['o_type'];
22        $oPrice = $row['o_price'];
23        $oQuantity = $row['o_quantity'];
24    }
25
26    //To give new values to DATABASE
27    if(isset($_POST['update']))
28    {
29        //Now given Variable name || Textbox id name
30        $ID = $_POST['oth_no'];
31        $Name = $_POST['oth_name'];
32        $Type = $_POST['oth_type'];
33        $Price = $_POST['oth_price'];
34        $Quantity = $_POST['oth_quantity'];
35
36        // SQL query to update the data in user table where the id = $userid
37        // (Database column id name || Previously given Variable name)
38        $query = "UPDATE other SET o_card = '{$ID}' , o_name = '{$Name}' , o_type = '{$Type}' , o_price = '{$Price}' ,
39        o_quantity = '{$Quantity}' WHERE id = $userid";
40        $update_user = mysqli_query($conn, $query);
41        echo "<script type='text/javascript'>alert('Item edited successfully!')</script>";
42        header("location:../Other.php");
43    }
44
45    <h1 class="text-center">Edit Item Details</h1>
46    <div class="container ">
47        <form action="" method="post">
48
49            <div class="form-row">
50                <div class="form-group col-md-6">
51                    <label for="id">ID Number</label>

```

Figure 3. 155 Coding for Edit_Other.php part 1



```
52     <input type="text" class="form-control" name="oth_no" value="php echo $oId ?" maxlength="12" required>
53   </div>
54   <div class="form-group col-md-6">
55     <label for="name">Item Name</label>
56     <input type="text" class="form-control" name="oth_name" value="php echo $oName ?" maxlength="10" required>
57   </div>
58 </div>
59
60 <div class="form-row">
61   <div class="form-group col-md-6">
62     <label for="type">Type</label>
63     <select id="inputState" name="oth_type" class="form-control" value="php echo $oType ?">
64       <option selected>Office Stationary</option>
65       <option>Machines</option>
66       <option>Beds and Metresses</option>
67       <option>Vehicles</option>
68       <option>Other</option>
69     </select>
70   </div>
71   <div class="form-group col-md-6">
72     <label for="price">Price</label>
73     <input type="text" class="form-control" name="oth_price" value="php echo $oPrice ?">
74   </div>
75 </div>
76
77 <div class="form-row">
78   <div class="form-group col-md-12">
79     <label for="quantity">Quantity</label>
80     <input type="text" class="form-control" name="oth_quantity" value="php echo $oQuantity ?">
81   </div>
82 </div>
83
84
85
86
87   <div class="col text-center">
88     <input type="submit" name="update" class="btn btn-info btn-large" value="Edit">
89   </div>
90
91 </form>
92 </div>
93
94 <!-- BACK button to go to the Other page -->
95 <div class="container text-center mt-5">
96   <a href="../Other.php" class="btn btn-warning mt-5"> Back </a>
97 </div>
98
99
100 <!-- Footer -->
101 <?php include "../../footer.php" ?>
```

Figure 3. 156 Coding for Edit_Other.php part 2

Other Functions

db.php :

Database linking codes

```

1 <?php
2 //server with default setting (user 'root' with no password)
3 $host = 'localhost'; // server
4 $user = 'root';
5 $pass = "";
6 $database = 'php_crud'; //Database Name
7
8 // establishing connection
9 $conn = mysqli_connect($host,$user,$pass,$database);
10
11 // for displaying an error msg in case the connection is not established
12 if (!$conn) {
13     die("Connection failed: " . mysqli_connect_error());
14 }
15 ?>
16
17
18

```

Figure 3. 157 Coding for db.php

header.php :

The header part of every php files I've created. This part is linked in above every php files as I mentioned in comments.

```

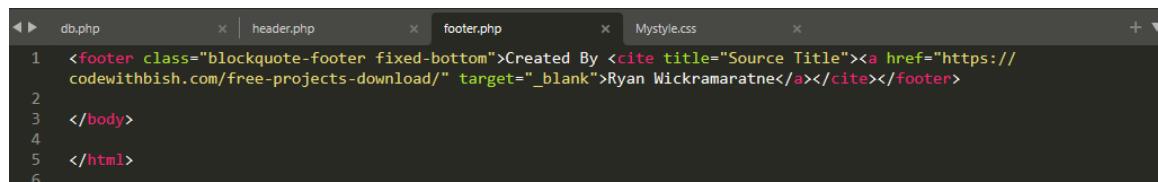
1 <?php include "db.php" ?>
2 <!DOCTYPE html>
3 <html lang="en">
4
5 <head>
6
7     <meta charset="UTF-8">
8     <meta http-equiv="X-UA-Compatible" content="IE=edge">
9     <meta name="viewport" content="width=device-width, initial-scale=1.0">
10
11     <!-- Bootstrap CSS -->
12     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet"
13          integrity="sha384-eOJMYsd53ii+sc0/bJFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6" crossorigin="anonymous">
14
15
16     <!-- Bootstrap CSS ->CDN Link -->
17     <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.3.0/font/bootstrap-icons.css">
18     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">
19
20     <!-- Bootstrap Icon -->
21     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
22
23     <!-- J Query Link with Popper and Stackpath-->
24     <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
25     <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"></script>
26     <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></script>
27
28     </head>
29
30     <body>

```

Figure 3. 158 Coding for header.php

footer.php :

The footer part of every php files I've created. This part is linked in almost every php files as I mentioned in comments.

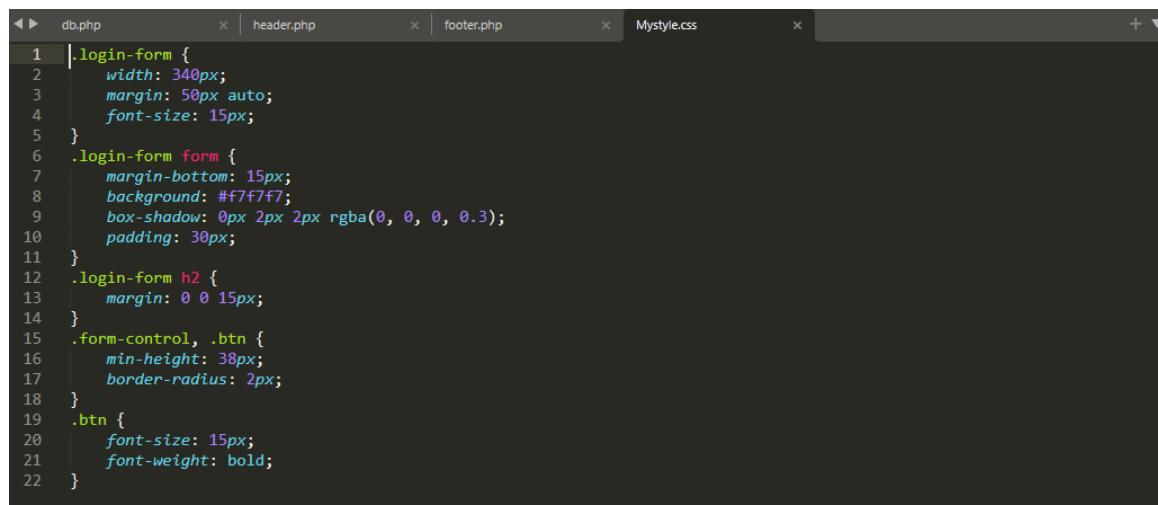


```
1 <footer class="blockquote-footer fixed-bottom">Created By <cite title="Source Title"><a href="https://
2 codewithbish.com/free-projects-download/" target="_blank">Ryan Wickramaratne</a></cite></footer>
3
4 </body>
5
6 </html>
```

Figure 3. 159 Coding for footer.php

Mystyle.css:

The CSS part I've created for every php files. This part is linked in header.php file so above every php files can inherit this CSS file.



```
1 .login-form {
2     width: 340px;
3     margin: 50px auto;
4     font-size: 15px;
5 }
6 .login-form form {
7     margin-bottom: 15px;
8     background: #f7f7f7;
9     box-shadow: 0px 2px 2px rgba(0, 0, 0, 0.3);
10    padding: 30px;
11 }
12 .login-form h2 {
13     margin: 0 0 15px;
14 }
15 .form-control, .btn {
16     min-height: 38px;
17     border-radius: 2px;
18 }
19 .btn {
20     font-size: 15px;
21     font-weight: bold;
22 }
```

Figure 3. 160 Coding for Mystyle.css

3.3.4 Final product's File Organization in WAMP Server

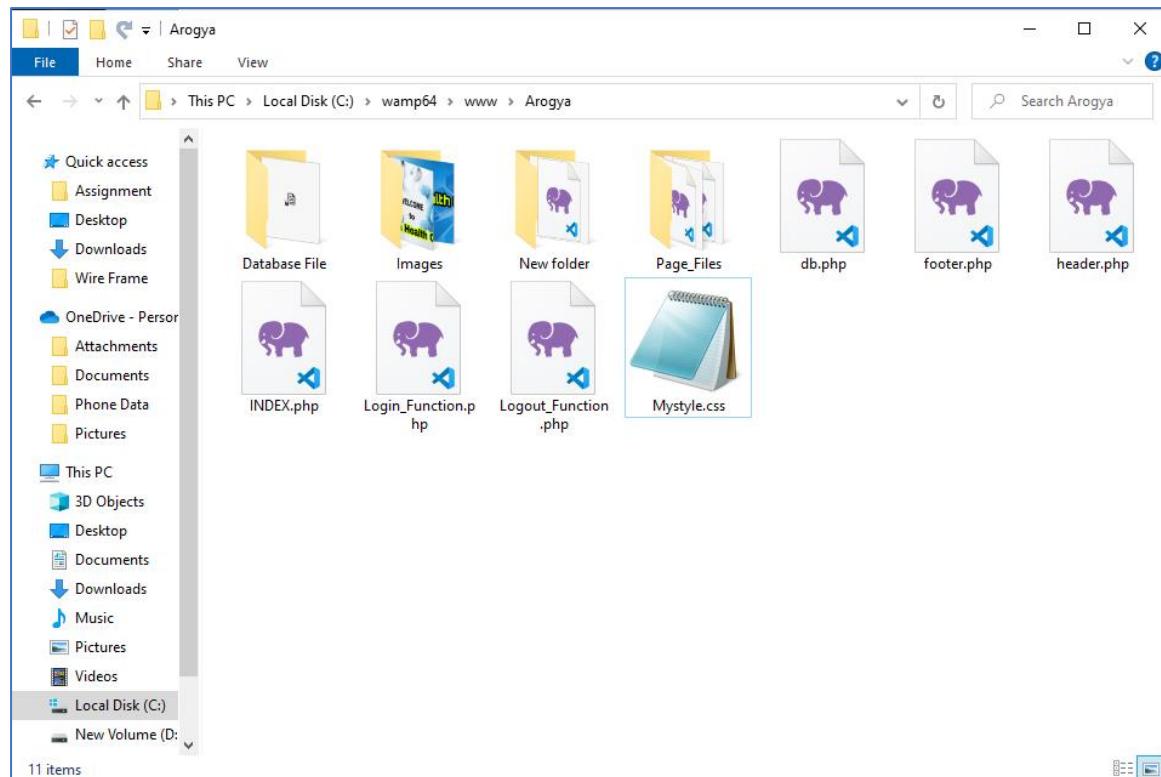


Figure 3. 161 Final product's File Organization in WAMP Server part 1

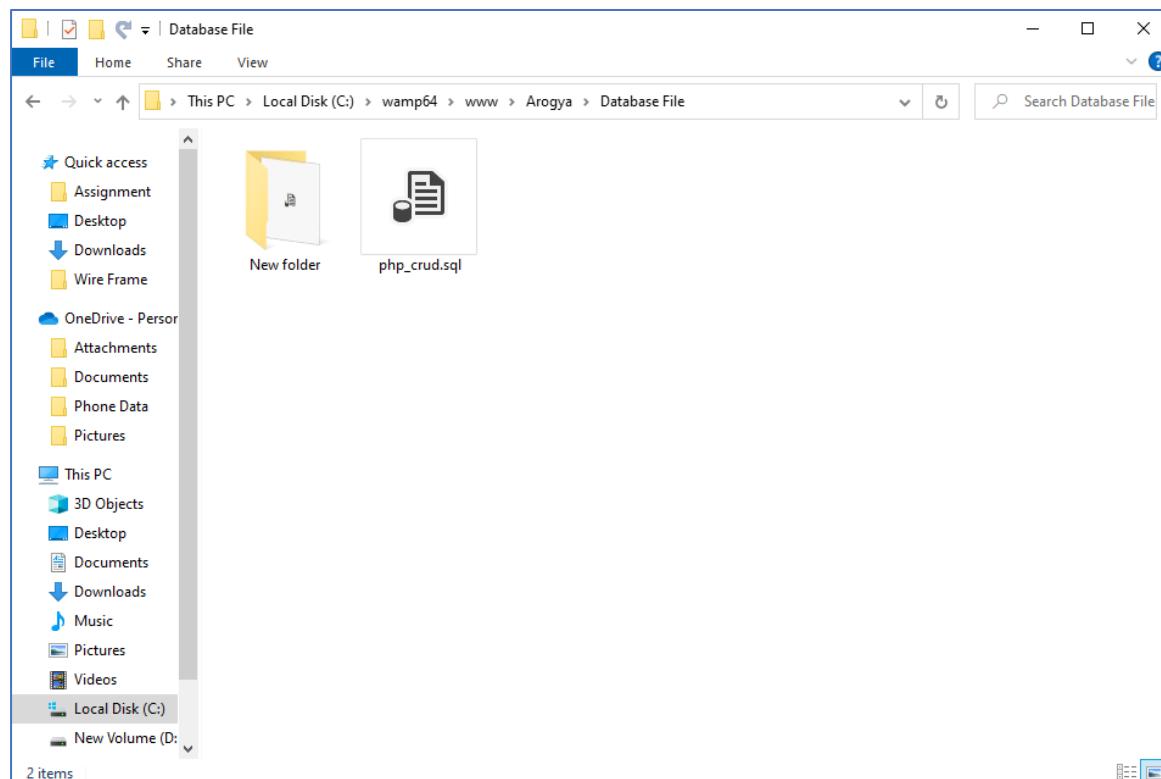


Figure 3. 162 Final product's File Organization in WAMP Server part 2

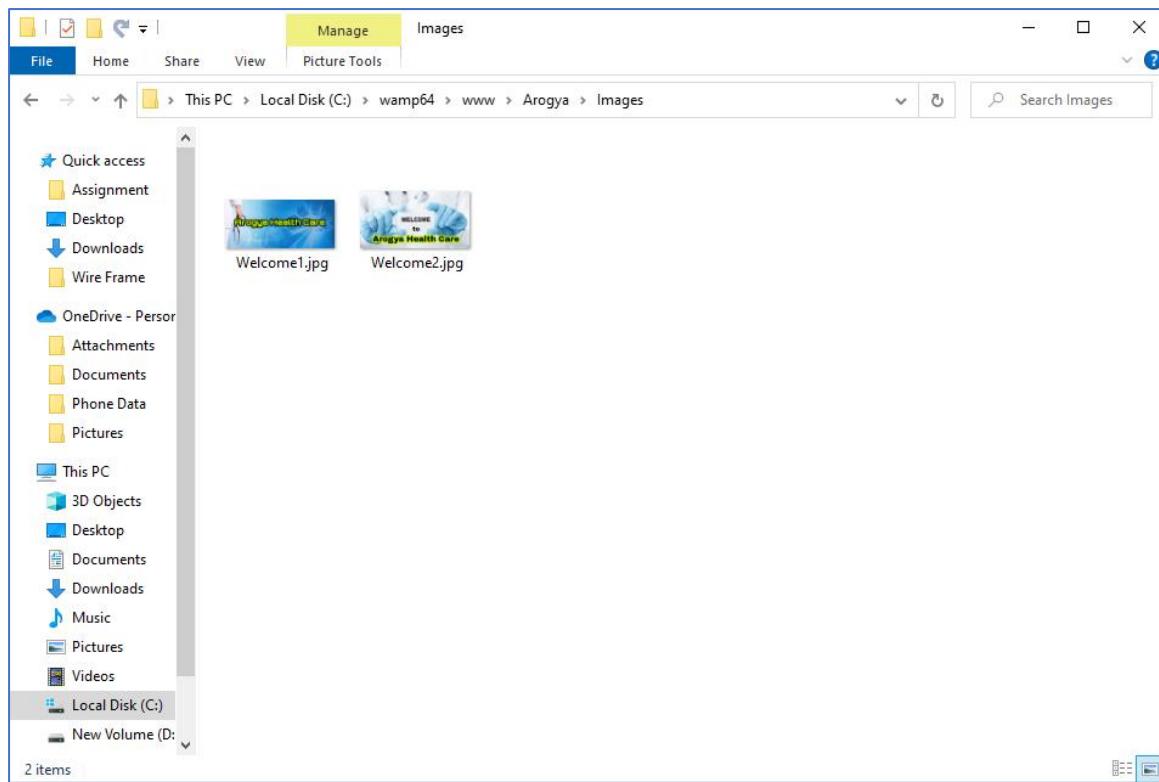


Figure 3. 163 Final product's File Organization in WAMP Server part 3

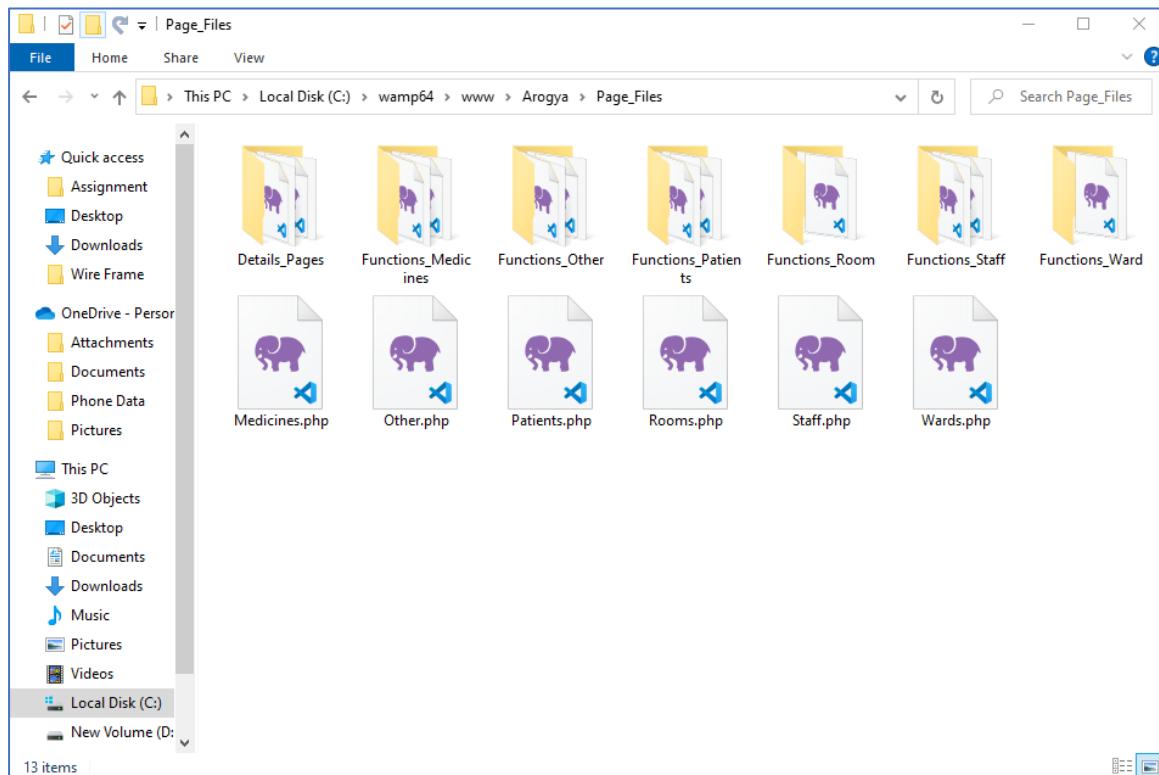


Figure 3. 164 Final product's File Organization in WAMP Server part 4

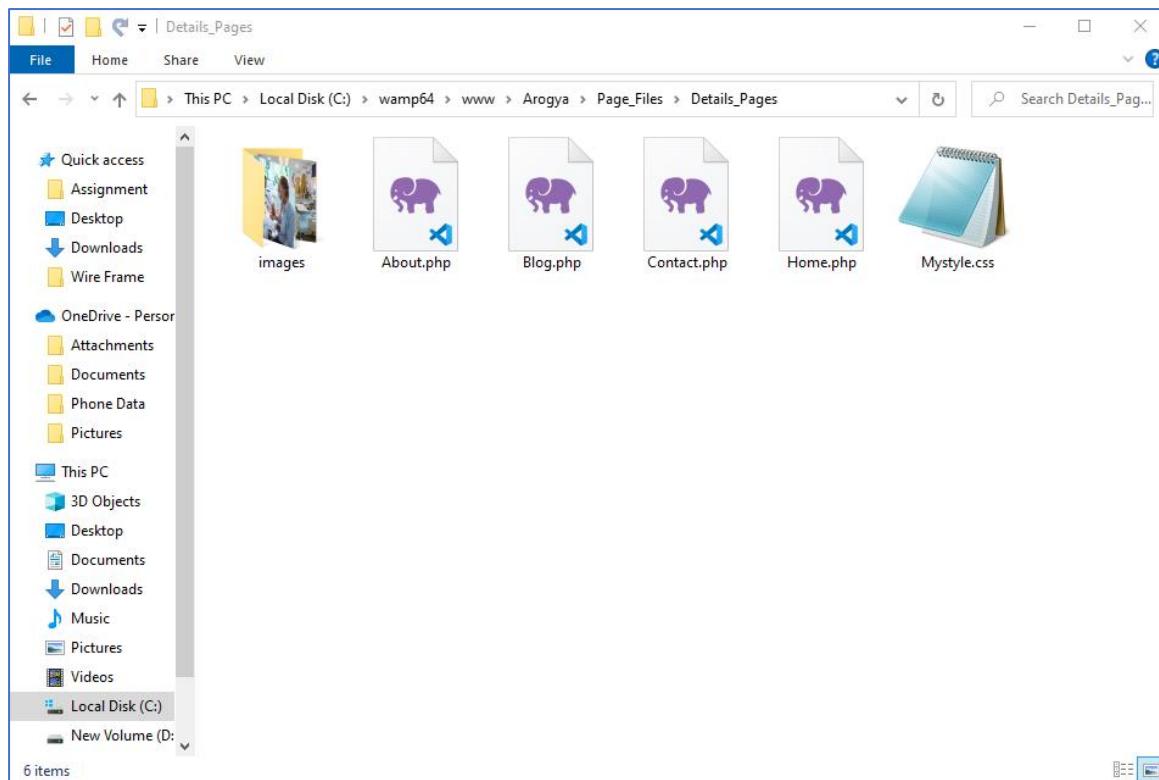


Figure 3. 165 Final product's File Organization in WAMP Server part 5

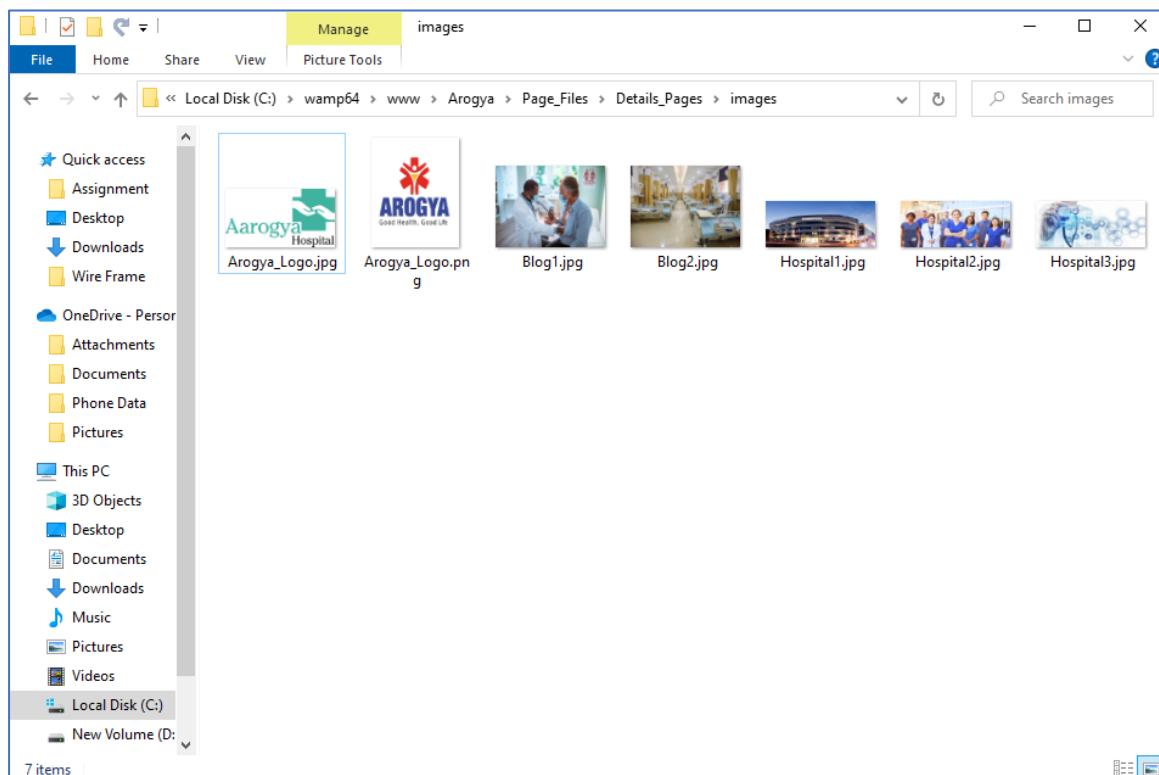


Figure 3. 166 Final product's File Organization in WAMP Server part 6

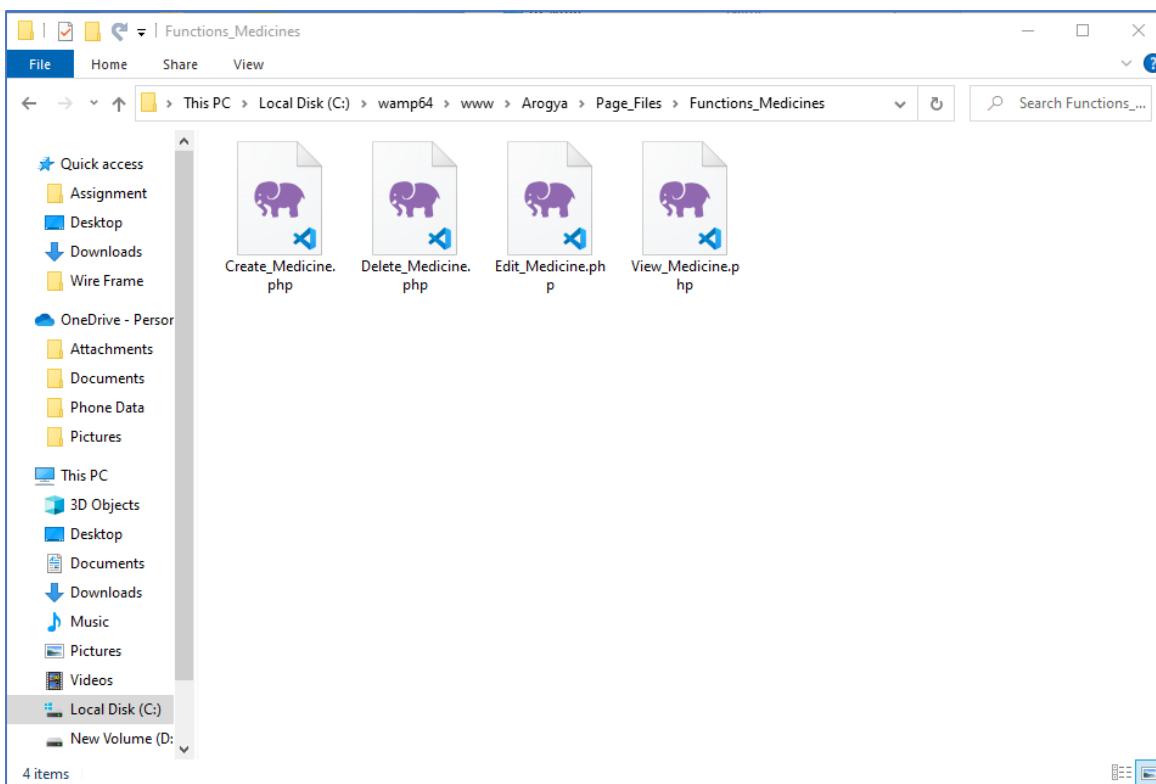


Figure 3. 167 Final product's File Organization in WAMP Server part 7

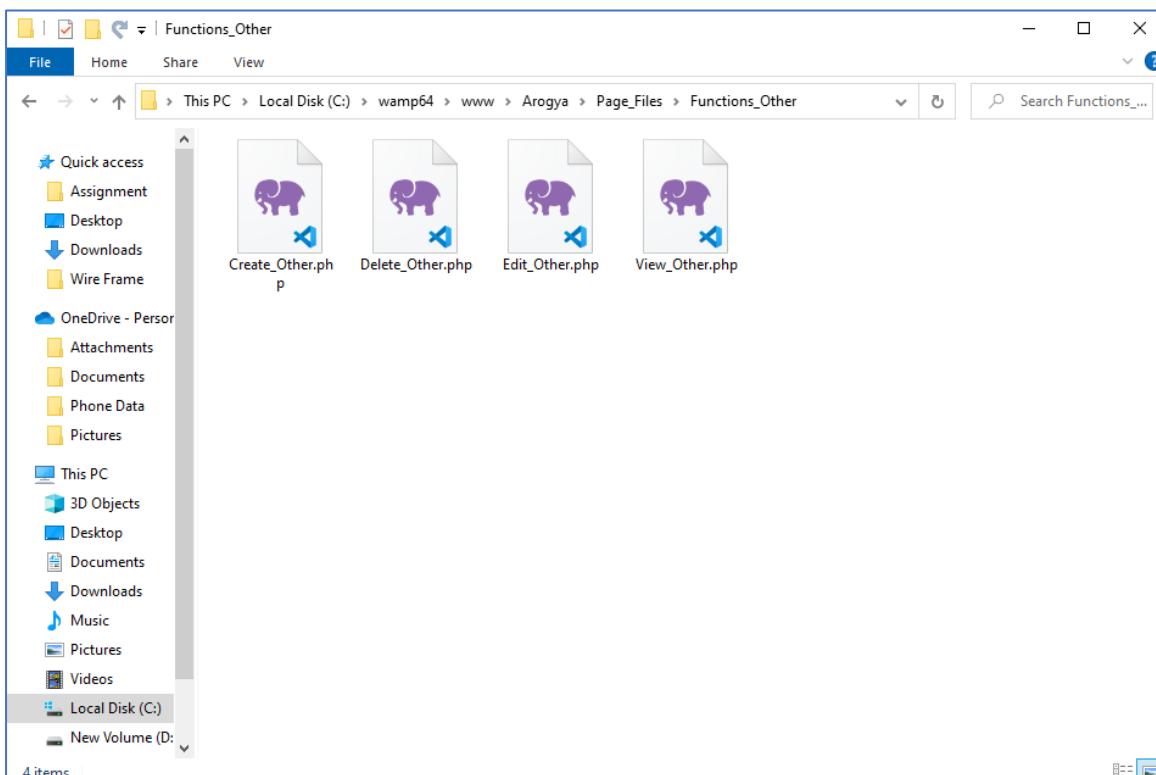


Figure 3. 168 Final product's File Organization in WAMP Server part 8

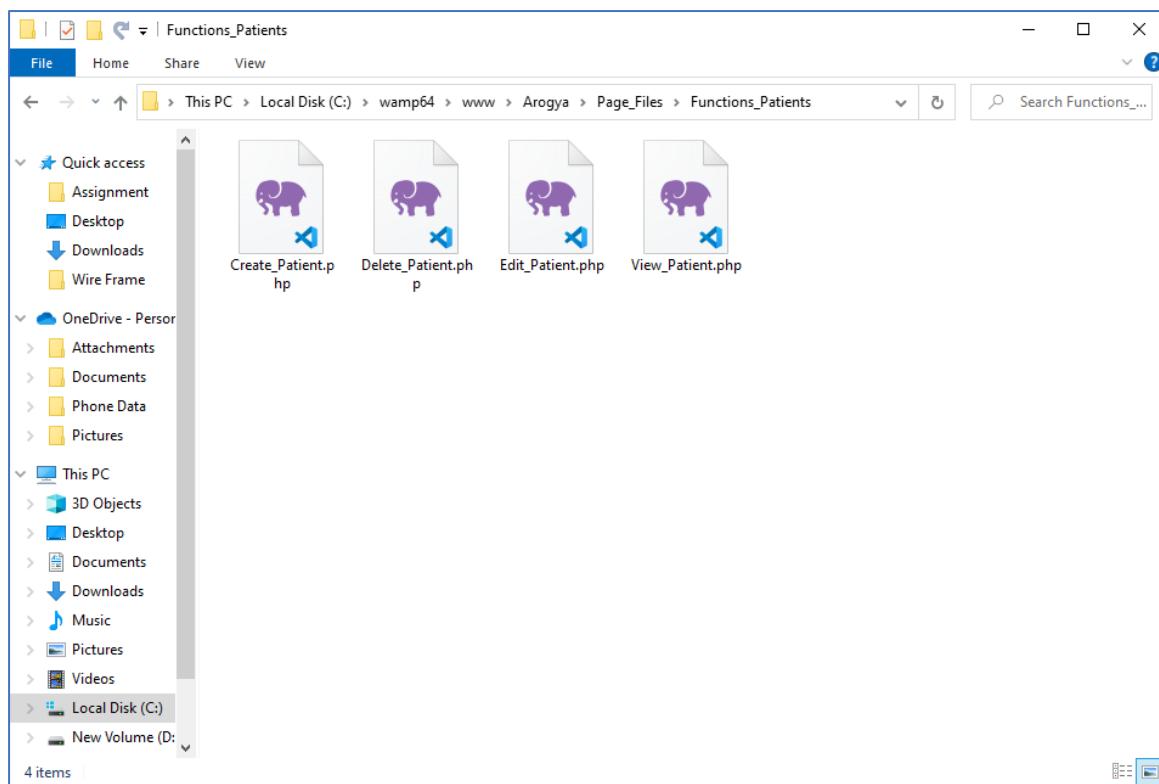


Figure 3. 169 Final product's File Organization in WAMP Server part 9

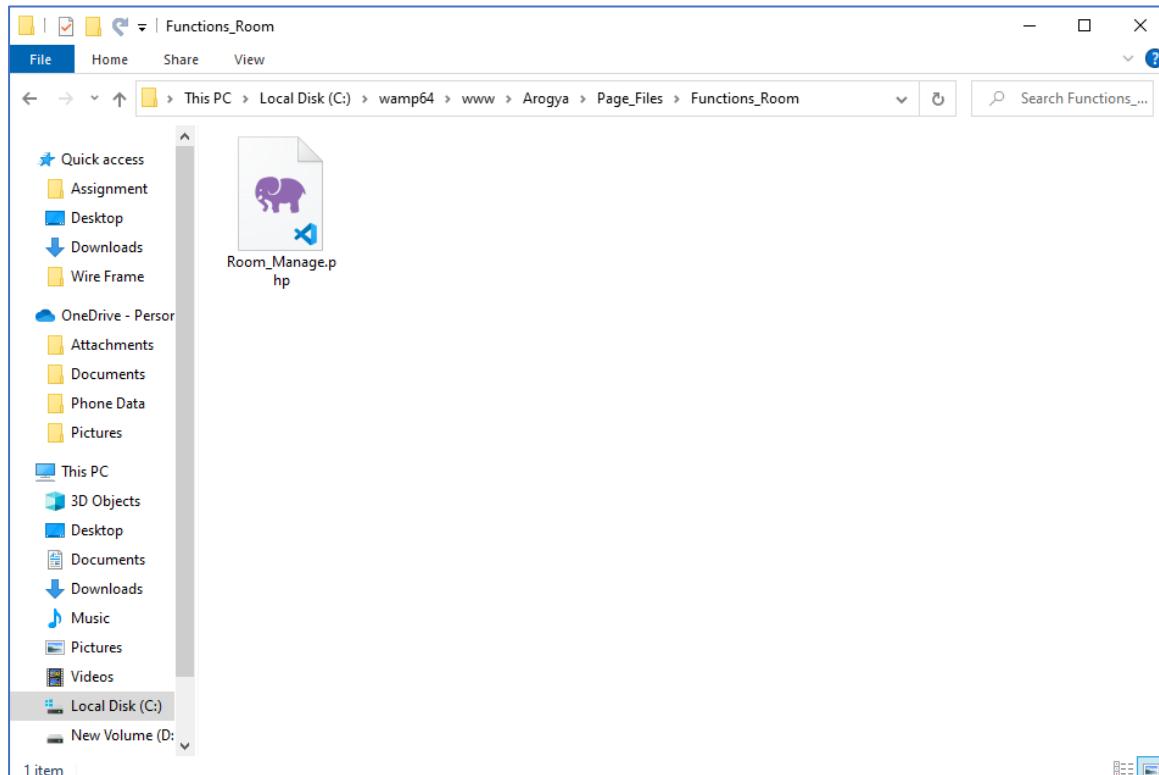


Figure 3. 170 Final product's File Organization in WAMP Server part 10

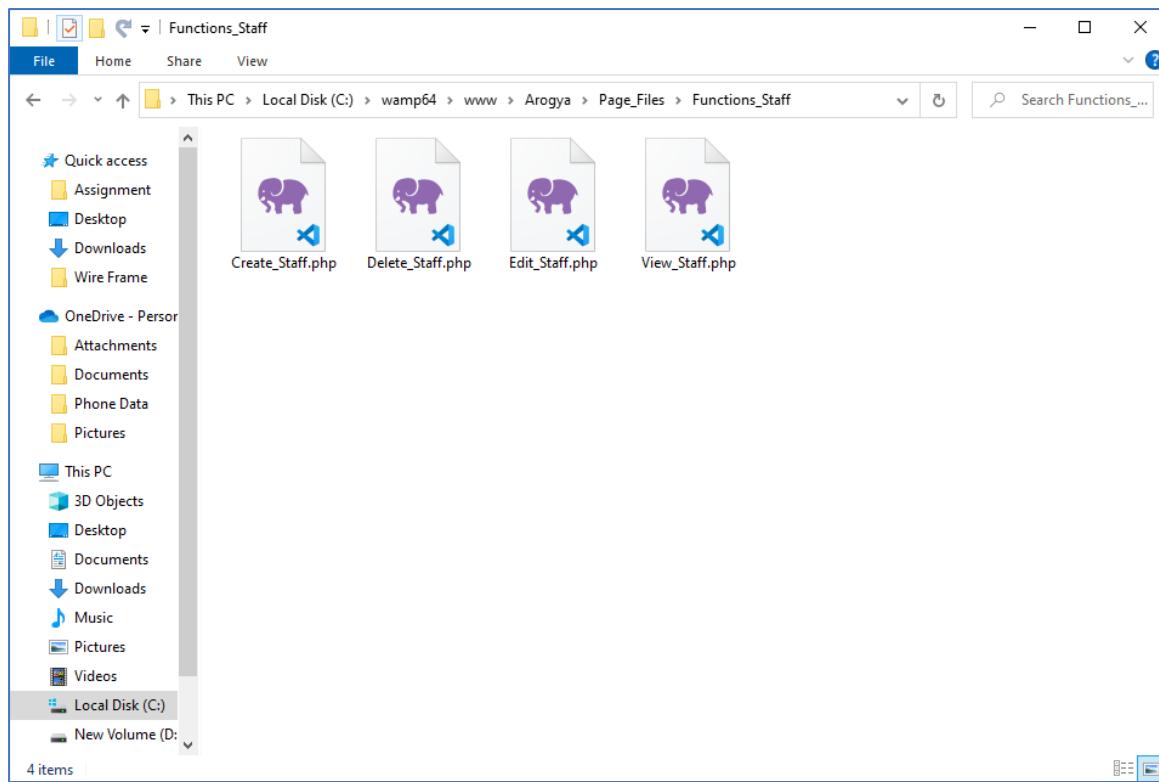


Figure 3. 171 Final product's File Organization in WAMP Server part 11

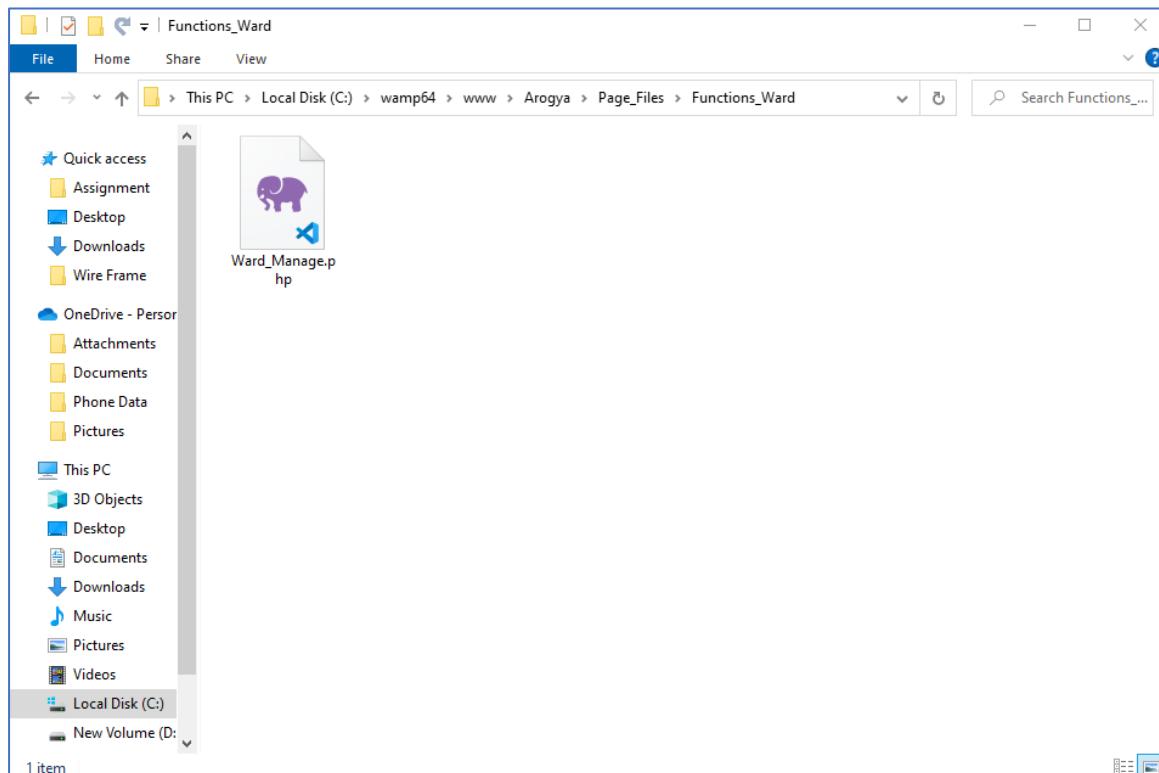


Figure 3. 172 Final product's File Organization in WAMP Server part 12

3.3.5 Final product's Database interfaces

Interfaces

MySQL Database (“php_crud” database) :

The screenshot shows the phpMyAdmin interface for the 'php_crud' database. The left sidebar lists databases: information_schema, mysql, performance_schema, and the current database, php_crud, which contains tables: admin, bills, medicine, other, patient, rooms, staff, and wards. The main panel displays the 'Structure' tab for the 'php_crud' database, showing a list of 8 tables. A table summary at the bottom indicates 34 rows and 0 B overhead.

| Table | Action | Rows | Type | Collation | Size | Overhead |
|----------|---|------|--------|--------------------|----------|----------|
| admin | Browse Structure Search Insert Empty Drop | 2 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |
| bills | Browse Structure Search Insert Empty Drop | 5 | MyISAM | latin1_swedish_ci | 2.1 KiB | - |
| medicine | Browse Structure Search Insert Empty Drop | 4 | MyISAM | latin1_swedish_ci | 2.2 KiB | - |
| other | Browse Structure Search Insert Empty Drop | 4 | MyISAM | latin1_swedish_ci | 2.2 KiB | - |
| patient | Browse Structure Search Insert Empty Drop | 4 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |
| rooms | Browse Structure Search Insert Empty Drop | 4 | MyISAM | latin1_swedish_ci | 2.3 KiB | - |
| staff | Browse Structure Search Insert Empty Drop | 7 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |
| wards | Browse Structure Search Insert Empty Drop | 4 | MyISAM | utf8mb4_general_ci | 16.0 KiB | - |

Figure 3. 173 Final product's Database interfaces [MySQL Database (“php_crud” database)]

Admin Table :

The screenshot shows the 'Table structure' tab for the 'admin' table. The table has three columns:

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|------------|-------------|--------------------|------------|------|---------|----------|----------------|--|
| 1 | id | int(10) | | | No | None | | AUTO_INCREMENT | Change Drop More |
| 2 | u_name | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 3 | u_password | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |

Indexes:

| Action | Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|--|---------|-------|--------|--------|--------|-------------|-----------|------|---------|
| Edit Rename Drop | PRIMARY | BTREE | Yes | No | id | 3 | A | No | |

Figure 3. 174 Final product's Database interfaces [Admin Table] – Table Structure

The screenshot shows the 'Structure' tab for the 'admin' table. It displays the following data:

| id | u_name | u_password |
|----|--------|------------|
| 1 | admin | admin |
| 3 | ryan | ryan |

Figure 3. 175 Final product's Database interfaces [Admin Table] – Inserted data

Bills Table :

The screenshot shows the phpMyAdmin interface for the 'bills' table in the 'php_crud' database. The table has five columns: id, bill_no, bill_name, bill_total, and bill_phone. The 'id' column is defined as int(11) with AUTO_INCREMENT, while the others are varchar(20) with utf8mb4_general_ci collation. A primary key is defined on the 'id' column. The 'bill_no' column is indexed.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|------------|-------------|--------------------|------------|------|---------|----------------|--------------------|--------------------|
| 1 | id | int(11) | utf8mb4_general_ci | | No | None | AUTO_INCREMENT | Change Drop More | Change Drop More |
| 2 | bill_no | varchar(20) | utf8mb4_general_ci | | No | None | | Change Drop More | Change Drop More |
| 3 | bill_name | varchar(20) | utf8mb4_general_ci | | No | None | | Change Drop More | Change Drop More |
| 4 | bill_total | varchar(20) | utf8mb4_general_ci | | No | None | | Change Drop More | Change Drop More |
| 5 | bill_phone | varchar(20) | utf8mb4_general_ci | | No | None | | Change Drop More | Change Drop More |

Figure 3. 176 Final product's Database interfaces [Bills Table] – Table Structure

The screenshot shows the phpMyAdmin interface displaying the data from the 'bills' table. Five rows are listed, each with columns: id, bill_no, bill_name, bill_total, and bill_phone. The data includes entries for Ryan, Judith, safd, Yupun, and fgdertg.

| | id | bill_no | bill_name | bill_total | bill_phone |
|---|----|---------|-----------|------------|------------|
| 1 | 1 | 1 | Ryan | 1000 | 0764170647 |
| 2 | 2 | 2 | Judith | 500 | 0764170485 |
| 3 | 3 | 3 | safd | 23 | 2132 |
| 4 | 4 | 4 | Yupun | 1000 | 524520 |
| 5 | 10 | 10 | fgdertg | 545 | 525235 |

Figure 3. 177 Final product's Database interfaces [Bills Table] – Inserted data

Medicine Table :

Table: medicine

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|------------|-------------|--------------------|------------|------|---------|----------------|-------|--|
| 1 | id | int(10) | utf8mb4_general_ci | | No | None | AUTO_INCREMENT | | Change Drop More |
| 2 | m_card | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 3 | m_name | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 4 | m_type | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 5 | m_price | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 6 | m_quantity | int(50) | | | No | None | | | Change Drop More |

Figure 3. 178 Final product's Database interfaces [Medicine Table] – Table Structure

Showing rows 0 - 3 (4 total, Query took 0.0002 seconds.)

```
SELECT * FROM `medicine`
```

| | id | m_card | m_name | m_type | m_price | m_quantity |
|---|----|--------------|--------------|--------|---------|------------|
| 1 | 1 | Panadol | Antibiotics | 5 | 100 | |
| 2 | 3 | Paracetamol | Pain Killers | 10 | 500 | |
| 3 | 4 | Cough Syrup | Other | 100 | 50 | |
| 4 | 5 | Gastic Syrup | Other | 100 | 50 | |

Figure 3. 179 Final product's Database interfaces [Medicine Table] – Inserted data

Other Table :

The screenshot shows the phpMyAdmin interface for the 'other' table in the 'php_crud' database. The table has six columns: id, o_card, o_name, o_type, o_price, and o_quantity. The 'id' column is defined as int(10) with AUTO_INCREMENT, while the others are varchar(50). The 'o_quantity' column is defined as int(50). There is one primary key index on the 'id' column.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|------------|-------------|--------------------|------------|------|---------|----------------|--|--------|
| 1 | id | int(10) | utf8mb4_general_ci | | No | None | AUTO_INCREMENT | Change Drop More | |
| 2 | o_card | varchar(50) | utf8mb4_general_ci | | No | None | | Change Drop More | |
| 3 | o_name | varchar(50) | utf8mb4_general_ci | | No | None | | Change Drop More | |
| 4 | o_type | varchar(50) | utf8mb4_general_ci | | No | None | | Change Drop More | |
| 5 | o_price | varchar(50) | utf8mb4_general_ci | | No | None | | Change Drop More | |
| 6 | o_quantity | int(50) | | | No | None | | Change Drop More | |

Figure 3. 180 Final product's Database interfaces [Other Table] – Table Structure

The screenshot shows the phpMyAdmin interface displaying the data inserted into the 'other' table. The table contains five rows of data:

| | id | o_card | o_name | o_type | o_price | o_quantity |
|---|----|--------|-----------|--------------------|---------|------------|
| 1 | 1 | 1 | Pen | Office Stationary | 10 | 20 |
| 2 | 3 | 2 | Bed | Beds and Metresses | 10000 | 5 |
| 3 | 4 | 3 | Metresses | Beds and Metresses | 5000 | 10 |
| 4 | 5 | 4 | Van | Vehicles | 4000000 | 2 |

Figure 3. 181 Final product's Database interfaces [Other Table] – Inserted data

Patient Table :

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|----|------------------------|---------------------------|---------------------------------|------------|------|-------------------|----------|-----------------------------|--------------------|
| 1 | <code>id</code> | <code>int(10)</code> | <code>utf8mb4_general_ci</code> | | No | <code>None</code> | | <code>AUTO_INCREMENT</code> | Change Drop More |
| 2 | <code>p_card</code> | <code>varchar(50)</code> | <code>utf8mb4_general_ci</code> | | No | <code>None</code> | | | Change Drop More |
| 3 | <code>p_f_name</code> | <code>varchar(50)</code> | <code>utf8mb4_general_ci</code> | | No | <code>None</code> | | | Change Drop More |
| 4 | <code>p_l_name</code> | <code>varchar(50)</code> | <code>utf8mb4_general_ci</code> | | No | <code>None</code> | | | Change Drop More |
| 5 | <code>p_address</code> | <code>varchar(50)</code> | <code>utf8mb4_general_ci</code> | | No | <code>None</code> | | | Change Drop More |
| 6 | <code>p_mobile</code> | <code>varchar(50)</code> | <code>utf8mb4_general_ci</code> | | No | <code>None</code> | | | Change Drop More |
| 7 | <code>p_gender</code> | <code>varchar(20)</code> | <code>utf8mb4_general_ci</code> | | No | <code>None</code> | | | Change Drop More |
| 8 | <code>p_age</code> | <code>int(10)</code> | | | No | <code>None</code> | | | Change Drop More |
| 9 | <code>p_admit</code> | <code>varchar(10)</code> | <code>utf8mb4_general_ci</code> | | No | <code>None</code> | | | Change Drop More |
| 10 | <code>p_history</code> | <code>varchar(255)</code> | <code>utf8mb4_general_ci</code> | | No | <code>None</code> | | | Change Drop More |

Figure 3. 182 Final product's Database interfaces [Patient Table] – Table Structure

| | <code>id</code> | <code>p_card</code> | <code>p_f_name</code> | <code>p_l_name</code> | <code>p_address</code> | <code>p_mobile</code> | <code>p_gender</code> | <code>p_age</code> | <code>p_admit</code> | <code>p_history</code> |
|--|-----------------|---------------------|-----------------------|-----------------------|-------------------------------------|-----------------------|-----------------------|--------------------|----------------------|---|
| | 7 | 1 | Judith | Michelle | test | 0764170485 | Male | 20 | Ward 2 | Good medical history. |
| | 8 | 2 | Ryan | Wickramaratne | 181/124, Baseline Road, Negombo. | 0764170647 | Male | 25 | Ward 1 | Well behaved |
| | 9 | 3 | Pavel | Fernando | 610 King Street Holyoke, MA 01040 | 0769845321 | Male | 15 | Ward 3 | 15 yo kidâ€“ swelling of tongue and difficulty bre... |
| | 10 | 4 | Malshan | Fernando | 105 Swanson St. Pataskala, OH 43062 | 0756419789 | Male | 25 | Ward 2 | 25 yo man â€“ new onset of fever, HTN, rigidity an... |

Figure 3. 183 Final product's Database interfaces [Patient Table] – Inserted data

Rooms Table :

Table: rooms

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---------|-------------|--------------------|------------|------|---------|----------------|-------|--|
| 1 | id | int(10) | | | No | None | AUTO_INCREMENT | | Change Drop More |
| 2 | ward_id | varchar(20) | utf8_general_ci | | No | None | | | Change Drop More |
| 3 | room1 | varchar(20) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 4 | room2 | varchar(20) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 5 | room3 | varchar(20) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 6 | room4 | varchar(20) | utf8mb4_general_ci | | No | None | | | Change Drop More |

Figure 3. 184 Final product's Database interfaces [Rooms Table] – Table Structure

Showing rows 0 - 3 (4 total, Query took 0.0004 seconds.)

| | id | ward_id | room1 | room2 | room3 | room4 |
|---|----|---------|-----------|-----------|-----------|-----------|
| 1 | 1 | Ward 1 | No | No | Available | Available |
| 2 | 2 | Ward 2 | Available | Available | Available | Available |
| 3 | 3 | Ward 3 | No | No | No | Available |
| 4 | 4 | Ward 4 | No | No | No | No |

Figure 3. 185 Final product's Database interfaces [Rooms Table] – Inserted data

Staff Table :

Table Structure

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---------------------------|-------------|--------------------|------------|------|---------|----------------|--|--------|
| 1 | id | int(10) | utf8mb4_general_ci | | No | None | AUTO_INCREMENT | Change Drop More | |
| 2 | s_card | varchar(10) | utf8mb4_general_ci | | No | None | | Change Drop More | |
| 3 | s_f_name | varchar(50) | utf8mb4_general_ci | | No | None | | Change Drop More | |
| 4 | s_l_name | varchar(50) | utf8mb4_general_ci | | No | None | | Change Drop More | |
| 5 | s_phone | varchar(20) | utf8mb4_general_ci | | No | None | | Change Drop More | |
| 6 | s_address | varchar(50) | utf8mb4_general_ci | | No | None | | Change Drop More | |
| 7 | s_gender | varchar(10) | utf8mb4_general_ci | | No | None | | Change Drop More | |
| 8 | s_type | varchar(10) | utf8mb4_general_ci | | No | None | | Change Drop More | |
| 9 | s_dob | varchar(50) | utf8mb4_general_ci | | No | None | | Change Drop More | |

Figure 3. 186 Final product's Database interfaces [Staff Table] – Table Structure

Showing rows 0 - 6 (7 total, Query took 0.0004 seconds.)

| | Edit | Copy | Delete | id | s_card | s_f_name | s_l_name | s_phone | s_address | s_gender | s_type | s_dob |
|----------------------|----------------------|------------------------|------------------------|--------------------|------------------------|--------------------------|--------------------------|---|---------------------------|--------------------------|------------------------|--------------------------|
| Edit | Copy | Delete | 13 | 1 | Pavel | Fernandoooo | 31222721111 | Basiyawatta | | Male | Doctor | 2022-05-14 |
| Edit | Copy | Delete | 20 | 2 | Ryan | Wickramaratne | 764170647 | Negombo | | Male | Nurse | 2022-06-07 |
| Edit | Copy | Delete | 27 | 3 | Judith | Michelle | 0764170647 | 8933 Lower River Lane Muncie, IN 47302 | | Male | Technician | 1997-06-24 |
| Edit | Copy | Delete | 28 | 4 | Kyle | Wickramaratne | 0764357567 | 36 Bitarwood St. Holly Springs, NC 27540 | | Male | Attendant | 2022-01-05 |
| Edit | Copy | Delete | 29 | 5 | Sadew | Fernando | 0745196753 | 105 Swanson St. Pataskala, OH 43062 | | Male | Doctor | 2022-06-07 |
| Edit | Copy | Delete | 30 | 6 | Mashi | Perera | 0749823584 | 7456 N. Wellington Drive Dalton, GA 30721 167 Katch Harbour Dr | | Female | Nurse | 2022-06-10 2022-06-10 |

Figure 3. 187 Final product's Database interfaces [Staff Table] – Inserted data

Wards Table :

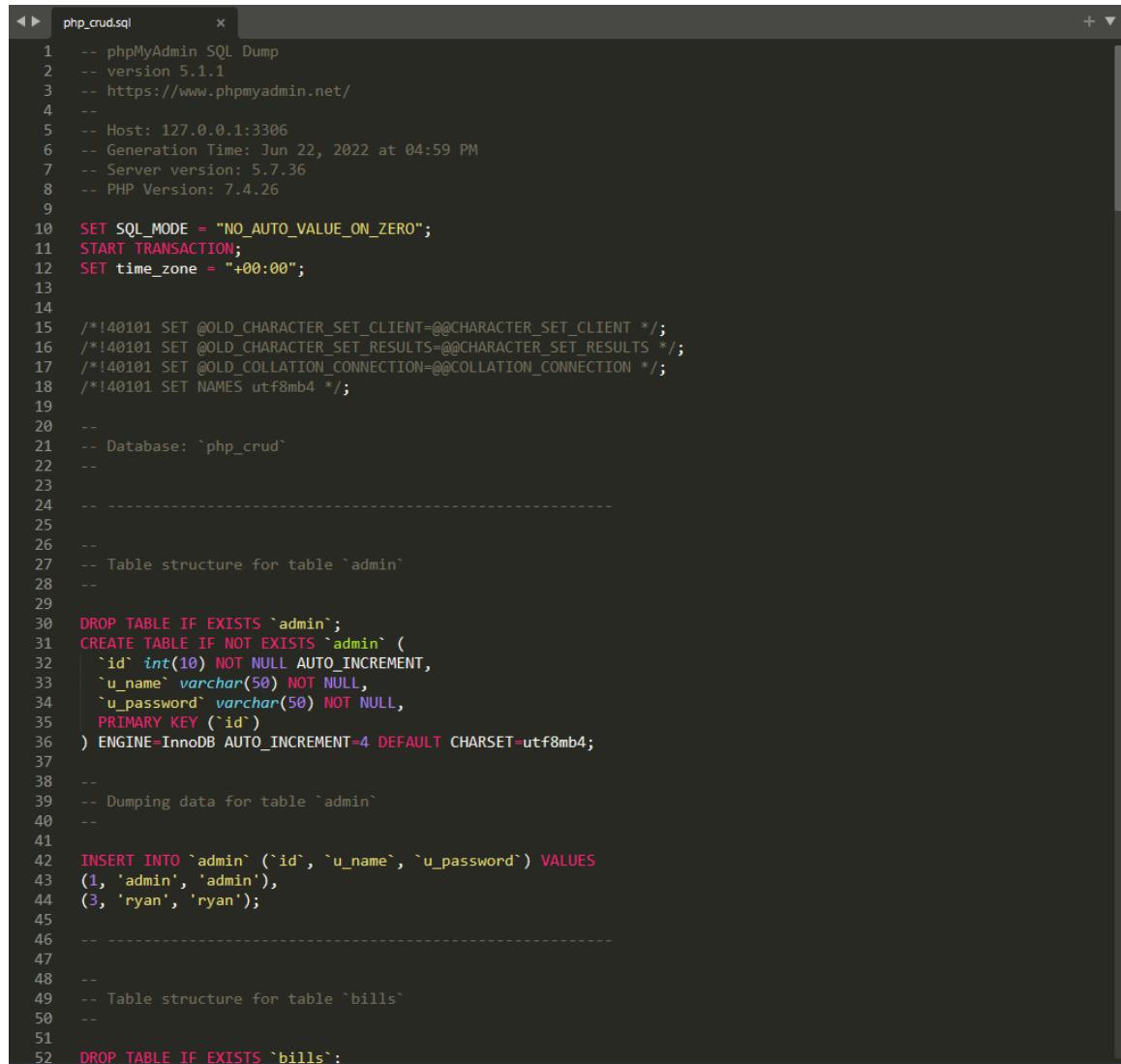
| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|----|-------------------|-------------|--------------------|------------|------|---------|----------------|-------|--|
| 1 | id | int(10) | utf8mb4_general_ci | | No | None | AUTO_INCREMENT | | Change Drop More |
| 2 | w_number | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 3 | Doctor1 | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 4 | Doctor2 | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 5 | Nurse1 | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 6 | Nurse2 | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 7 | Nurse3 | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 8 | Nurse4 | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 9 | Attendant1 | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 10 | Attendant2 | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |

Figure 3. 188 Final product's Database interfaces [Wards Table] – Table Structure

| | id | w_number | Doctor1 | Doctor2 | Nurse1 | Nurse2 | Nurse3 | Nurse4 | Attendant1 | Attendant2 |
|--|-----------|-----------------|----------------|----------------|--------------------|---------------|--------------------|---------------|-------------------|-------------------|
| | 1 | Ward 1 | Pavel Fernando | Sadew Fernando | Ryan Wickramaratne | Mashi Perera | Kyle Wickramaratne | Shyni Perera | | |
| | 2 | Ward 2 | Pavel Fernando | | Ryan Wickramaratne | | Kyle Wickramaratne | | | |
| | 3 | Ward 3 | Sadew Fernando | | Mashi Perera | | | | Shyni Perera | |
| | 4 | Ward 4 | Pavel Fernando | | Mashi Perera | | | | | |

Figure 3. 189 Final product's Database interfaces [Wards Table] – Inserted data

3.3.6 Final product's Database codes



```
php_crud.sql      x
1  -- phpMyAdmin SQL Dump
2  -- version 5.1.1
3  -- https://www.phpmyadmin.net/
4  --
5  -- Host: 127.0.0.1:3306
6  -- Generation Time: Jun 22, 2022 at 04:59 PM
7  -- Server version: 5.7.36
8  -- PHP Version: 7.4.26
9
10 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 START TRANSACTION;
12 SET time_zone = "+00:00";
13
14
15 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
16 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
17 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
18 /*!40101 SET NAMES utf8mb4 */;
19
20 --
21 -- Database: `php_crud`
22 --
23
24 --
25
26 --
27 -- Table structure for table `admin`
28 --
29
30 DROP TABLE IF EXISTS `admin`;
31 CREATE TABLE IF NOT EXISTS `admin` (
32     `id` int(10) NOT NULL AUTO_INCREMENT,
33     `u_name` varchar(50) NOT NULL,
34     `u_password` varchar(50) NOT NULL,
35     PRIMARY KEY (`id`)
36 ) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4;
37
38 --
39 -- Dumping data for table `admin`
40 --
41
42 INSERT INTO `admin` (`id`, `u_name`, `u_password`) VALUES
43 (1, 'admin', 'admin'),
44 (3, 'ryan', 'ryan');
45
46 --
47
48 --
49 -- Table structure for table `bills`
50 --
51
52 DROP TABLE IF EXISTS `bills`:
```

Figure 3. 190 Final product's Database codes part I

```

52 DROP TABLE IF EXISTS `bills`;
53 CREATE TABLE IF NOT EXISTS `bills` (
54     `id` int(11) NOT NULL AUTO_INCREMENT,
55     `bill_no` varchar(20) CHARACTER SET utf8mb4 NOT NULL,
56     `bill_name` varchar(20) CHARACTER SET utf8mb4 NOT NULL,
57     `bill_total` varchar(20) CHARACTER SET utf8mb4 NOT NULL,
58     `bill_phone` varchar(20) CHARACTER SET utf8mb4 NOT NULL,
59     PRIMARY KEY (`id`)
60 ) ENGINE=MyISAM AUTO_INCREMENT=6 DEFAULT CHARSET=latin1;
61
62 --
63 -- Dumping data for table `bills`
64 --
65
66 INSERT INTO `bills` (`id`, `bill_no`, `bill_name`, `bill_total`, `bill_phone`) VALUES
67 (1, '1', 'Ryan', '1000', '0764170647'),
68 (2, '2', 'Judith', '500', '0764170485'),
69 (3, '3', 'saafd', '23', '2132'),
70 (4, '4', 'Yupun', '1000', '524520'),
71 (5, '10', 'fgdertg', '545', '525235');
72
73 --
74
75 --
76 -- Table structure for table `medicine`
77 --
78
79 DROP TABLE IF EXISTS `medicine`;
80 CREATE TABLE IF NOT EXISTS `medicine` (
81     `id` int(10) NOT NULL AUTO_INCREMENT,
82     `m_card` varchar(50) CHARACTER SET utf8mb4 NOT NULL,
83     `m_name` varchar(50) CHARACTER SET utf8mb4 NOT NULL,
84     `m_type` varchar(50) CHARACTER SET utf8mb4 NOT NULL,
85     `m_price` varchar(50) CHARACTER SET utf8mb4 NOT NULL,
86     `m_quantity` int(50) NOT NULL,
87     PRIMARY KEY (`id`)
88 ) ENGINE=MyISAM AUTO_INCREMENT=6 DEFAULT CHARSET=latin1;
89
90 --
91 -- Dumping data for table `medicine`
92 --
93
94 INSERT INTO `medicine` (`id`, `m_card`, `m_name`, `m_type`, `m_price`, `m_quantity`) VALUES
95 (1, '1', 'Panadol', 'Antibiotics', '5', 100),
96 (3, '2', 'Paracetamo', 'Pain Killers', '10', 500),
97 (4, '3', 'Cough Syrup', 'Other', '100', 50),
98 (5, '4', 'Gastic Syrup', 'Other', '100', 50);
99
100 --
101
102 --
103 -- Table structure for table `other`

```

Figure 3. 191 Final product's Database codes part 2

```

php_crud.sql      x
103 -- Table structure for table `other`
104 --
105
106 DROP TABLE IF EXISTS `other`;
107 CREATE TABLE IF NOT EXISTS `other` (
108     `id` int(10) NOT NULL AUTO_INCREMENT,
109     `o_card` varchar(50) CHARACTER SET utf8mb4 NOT NULL,
110     `o_name` varchar(50) CHARACTER SET utf8mb4 NOT NULL,
111     `o_type` varchar(50) CHARACTER SET utf8mb4 NOT NULL,
112     `o_price` varchar(50) CHARACTER SET utf8mb4 NOT NULL,
113     `o_quantity` int(50) NOT NULL,
114     PRIMARY KEY (`id`)
115 ) ENGINE=MyISAM AUTO_INCREMENT=6 DEFAULT CHARSET=latin1;
116
117 --
118 -- Dumping data for table `other`
119 --
120
121 INSERT INTO `other`(`id`, `o_card`, `o_name`, `o_type`, `o_price`, `o_quantity`) VALUES
122 (1, '1', 'Pen', 'Office Stationary', '10', 20),
123 (3, '2', 'Bed', 'Beds and Metrresses', '10000', 5),
124 (4, '3', 'Metresses', 'Beds and Metrresses', '5000', 10),
125 (5, '4', 'Van', 'Vehicles', '4000000', 2);
126
127 -----
128
129 --
130 -- Table structure for table `patient`
131 --
132
133 DROP TABLE IF EXISTS `patient`;
134 CREATE TABLE IF NOT EXISTS `patient` (
135     `id` int(10) NOT NULL AUTO_INCREMENT,
136     `p_card` varchar(50) NOT NULL,
137     `p_f_name` varchar(50) NOT NULL,
138     `p_l_name` varchar(50) NOT NULL,
139     `p_address` varchar(50) NOT NULL,
140     `p_mobile` varchar(50) NOT NULL,
141     `p_gender` varchar(20) NOT NULL,
142     `p_age` int(10) NOT NULL,
143     `p_admit` varchar(10) NOT NULL,
144     `p_history` varchar(255) NOT NULL,
145     PRIMARY KEY (`id`)
146 ) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=utf8mb4;
147
148 --
149 -- Dumping data for table `patient`
150 --
151
152 INSERT INTO `patient`(`id`, `p_card`, `p_f_name`, `p_l_name`, `p_address`, `p_mobile`, `p_gender`, `p_age`, `p_admit`, `p_history`) VALUES
153 (7, '1', 'Judith', 'Michelle', 'test', '0764170485', 'Male', 20, 'Ward 2', 'Good medical history.');

```

Figure 3. 192 Final product's Database codes part 3

```

153 (7, '1', 'Judith', 'Michelle', 'test', '0764170485', 'Male', 20, 'Ward 2', 'Good medical history.'),
154 (8, '2', 'Ryan', 'Wickramaratne', '181/124, Baseline Road, Negombo.', '0764170647', 'Male', 25, 'Ward 1', 'Well
behaved'),
155 (9, '3', 'Pavel', 'Fernando', '610 King Street Holyoke, MA 01040', '0769845321', 'Male', 15, 'Ward 3', '15 yo
kidâ€“ swelling of tongue and difficulty breathing and swallowing'),
156 (10, '4', 'Malshan', 'Fernando', '105 Swanson St. Pataskala, OH 43062', '0756419789', 'Male', 25, 'Ward 2', '25
yo man â€“ new onset of fever, HTN, rigidity and altered mental status');

157
158 -- -----
159
160 --
161 -- Table structure for table `rooms`
162 --
163
164 DROP TABLE IF EXISTS `rooms`;
165 CREATE TABLE IF NOT EXISTS `rooms` (
166   `id` int(10) NOT NULL AUTO_INCREMENT,
167   `ward_id` varchar(20) CHARACTER SET utf8mb4 NOT NULL,
168   `room1` varchar(20) CHARACTER SET utf8mb4 NOT NULL,
169   `room2` varchar(20) CHARACTER SET utf8mb4 NOT NULL,
170   `room3` varchar(20) CHARACTER SET utf8mb4 NOT NULL,
171   `room4` varchar(20) CHARACTER SET utf8mb4 NOT NULL,
172   PRIMARY KEY (`id`)
173 ) ENGINE=MyISAM AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;
174
175 --
176 -- Dumping data for table `rooms`
177 --
178
179 INSERT INTO `rooms` (`id`, `ward_id`, `room1`, `room2`, `room3`, `room4`) VALUES
180 (1, 'Ward 1', 'No', 'No', 'Available', 'Available'),
181 (2, 'Ward 2', 'Available', 'Available', 'Available', 'Available'),
182 (3, 'Ward 3', 'No', 'No', 'No', 'Available'),
183 (4, 'Ward 4', 'No', 'No', 'No', 'No');
184
185 --
186
187 --
188 -- Table structure for table `staff`
189 --
190
191 DROP TABLE IF EXISTS `staff`;
192 CREATE TABLE IF NOT EXISTS `staff` (
193   `id` int(10) NOT NULL AUTO_INCREMENT,
194   `s_card` varchar(10) NOT NULL,
195   `s_f_name` varchar(50) NOT NULL,
196   `s_l_name` varchar(50) NOT NULL,
197   `s_phone` varchar(20) NOT NULL,
198   `s_address` varchar(50) NOT NULL,
199   `s_gender` varchar(10) NOT NULL,
200   `s_type` varchar(10) NOT NULL,
201   `s_dob` varchar(50) NOT NULL,

```

Figure 3. 193 Final product's Database codes part 4

```

1 `s_dob` varchar(50) NOT NULL,
2 PRIMARY KEY (`id`)
3 ) ENGINE=InnoDB AUTO_INCREMENT=32 DEFAULT CHARSET=utf8mb4;
4
5 --
6 -- Dumping data for table `staff`
7 --
8
9 INSERT INTO `staff` (`id`, `s_card`, `s_f_name`, `s_l_name`, `s_phone`, `s_address`, `s_gender`, `s_type`, `s_dob`)
10    VALUES
11 (13, '1', 'Pavel', 'Fernandoooo', '31222721111', 'Basiyawatta', 'Male', 'Doctor', '2022-05-14'),
12 (20, '2', 'Ryan', 'Wickramaratne', '764170647', 'Negombo', 'Male', 'Nurse', '2022-06-07'),
13 (27, '3', 'Judith', 'Michelle', '0764170647', '8933 Lower River Lane Muncie, IN 47302', 'Male', 'Technician', '1997-06-24'),
14 (28, '4', 'Kyle', 'Wickramaratne', '0764357567', '36 Briarwood St. Holly Springs, NC 27540', 'Male', 'Attendant', '2022-01-05'),
15 (29, '5', 'Sadew', 'Fernando', '0745196753', '105 Swanson St. Pataskala, OH 43062', 'Male', 'Doctor', '2022-06-07'),
16 (30, '6', 'Mashi ', 'Perera', '0749823584', '7456 N. Wellington Drive Dalton, GA 30721', 'Female', 'Nurse', '2022-06-10'),
17 (31, '7', 'Shyni', 'Perera', '0786431987', '162 Ketch Harbour Rd. Davenport, IA 52804', 'Female', 'Attendant', '2022-06-07');
18
19 --
20 --
21 -- Table structure for table `wards`
22 --
23
24 DROP TABLE IF EXISTS `wards`;
25 CREATE TABLE IF NOT EXISTS `wards` (
26     `id` int(10) NOT NULL AUTO_INCREMENT,
27     `w_number` varchar(50) NOT NULL,
28     `Doctor1` varchar(50) NOT NULL,
29     `Doctor2` varchar(50) NOT NULL,
30     `Nurse1` varchar(50) NOT NULL,
31     `Nurse2` varchar(50) NOT NULL,
32     `Nurse3` varchar(50) NOT NULL,
33     `Nurse4` varchar(50) NOT NULL,
34     `Attendant1` varchar(50) NOT NULL,
35     `Attendant2` varchar(50) NOT NULL,
36 PRIMARY KEY (`id`)
37 ) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4;
38
39 --
40 -- Dumping data for table `wards`
41 --
42
43 INSERT INTO `wards` (`id`, `w_number`, `Doctor1`, `Doctor2`, `Nurse1`, `Nurse2`, `Nurse3`, `Nurse4`, `Attendant1`, `Attendant2`) VALUES
44 (1, 'Ward 1', 'Pavel Fernandoooo', 'Sadew Fernando', 'Ryan Wickramaratne', 'Mashi Perera', '', '', 'Kyle Wickramaratne', 'Shyni Perera').

```

Figure 3. 194 Final product's Database codes part 5

```

243 INSERT INTO `wards` (`id`, `w_number`, `Doctor1`, `Doctor2`, `Nurse1`, `Nurse2`, `Nurse3`, `Nurse4`, `Attendant1`, `Attendant2`) VALUES
244 (1, 'Ward 1', 'Pavel Fernandoooo', 'Sadew Fernando', 'Ryan Wickramaratne', 'Mashi Perera', '', '', 'Kyle Wickramaratne', 'Shyni Perera'),
245 (2, 'Ward 2', 'Pavel Fernandoooo', '', 'Ryan Wickramaratne', '', '', '', 'Kyle Wickramaratne', ''),
246 (3, 'Ward 3', 'Sadew Fernando', '', 'Mashi Perera', '', '', '', 'Shyni Perera', ''),
247 (4, 'Ward 4', 'Pavel Fernandoooo', '', 'Mashi Perera', '', '', '', '', ''),
248 COMMIT;
249
250 /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
251 /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
252 /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
253

```

Figure 3. 195 Final product's Database codes part 6

3.4 Compare and Contrast the multipage website created (Final Output) to the Designed Document(Wireframe)

3.4.1 Overall Difference

It is obvious from the wireframes that this is not the finished design. A wireframe is not the same as an application's final appearance and feel. Few colours and low fidelity require us to prioritize structure over details. Once the construction is complete, there will be plenty of time for graphic design. Hence, when considering my wireframes and final outputs, the designs are the same look, but the final outputs contain sophisticated details and colors.

Wireframes can be created quite quickly; it probably only took a few minutes to create each page. The final ease of use is what matters most at this time, thus going through a few iterations is typical and expected. Hence, it took me a little time to consider the design and code the final output. By looking at these wireframe designs, I was able to make the output pages easily.

It is obvious from the wireframes that no code has yet been created. Instead of receiving a wireframe, a client or stakeholder would assume that they were receiving screenshots of the final product because the code behind those screenshots had already been built. Most of the time, this is not the case. This risk does not exist with wireframes. Hence, my wireframes had no codes within them. I was able to show it to my client (Arogya Hospital Management) and then get approval to create the product. Wireframes are the best way to start a project without hesitation or risk.

Table 3. 3 Overall Difference between multipage website created (Final Output) to the Designed Document(Wireframe)

| Final Output | Wireframe (Initial Design) |
|---|---|
| Interactive design to communicate the high-level concept of new product capabilities. | A brief illustration that conveys the broad idea of new product capabilities. |
| Designed to use and experience to client. | Designed to get understanding and gather internal feedback from the client. |
| Button is functional. | Button is non-functional (Only representing buttons). |
| Design fidelity is high. | Design fidelity is low. |
| Contains navigation and interactive components. | It includes a content format and structure. |
| Invested time is high. | Invested time is low. |
| Created by using WAMP server and some bootstrap coding. | Created through https://wireframe.cc/ |

3.4.2 Compare and Contrast each final output's pages with the initial design wireframe

Login Page

Table 3. 4 Login Page Difference between multipage website created (Final Output) to the Designed Document(Wireframe)

| Final Output | Wireframe (Initial Design) |
|---------------------------|---|
| Contains text. | No text (Lines represent text). |
| Contains actual pictures. | No actual picture (Box with cross represent picture). |
| Button is functional. | Button is non-functional (Only representing buttons). |
| Rich with colors. | No colors. |
| Placeholder available. | No placeholder. |

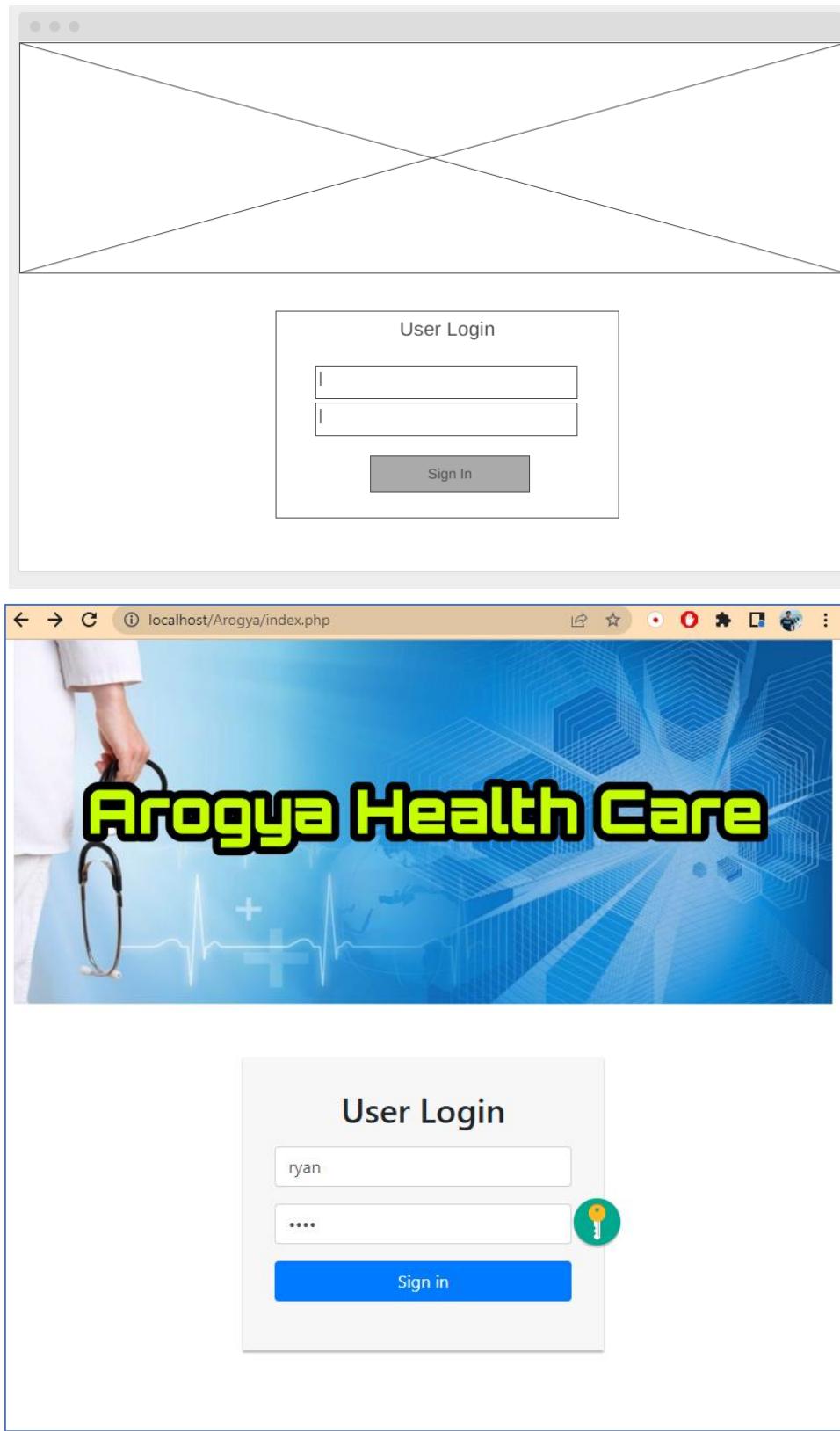


Figure 3. 196 Login Page Difference between multipage website created (Final Output) to the Designed Document(Wireframe)

Home Page

Table 3. 5 Home Page Difference between multipage website created (Final Output) to the Designed Document(Wireframe)

| Final Output | Wireframe (Initial Design) |
|---------------------------------|---|
| Contains text. | No text (Lines represent text). |
| Contains actual pictures. | No actual picture (Box with cross represent picture). |
| Only 1 picture is representing. | Carousel has 3 pictures with descriptions. |
| Buttons are functional. | Buttons are non-functional (Only representing buttons). |
| Rich with colors. | No colors. |
| Has logo. | Footer has no logo. |

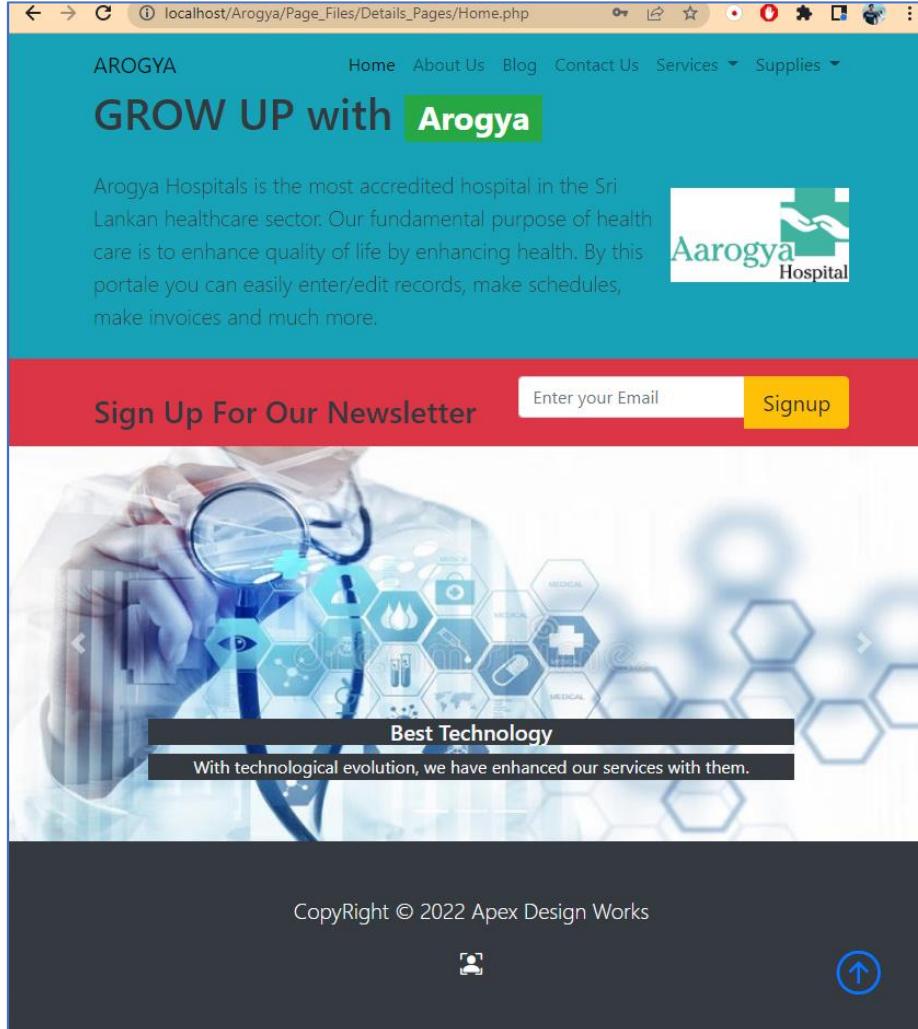
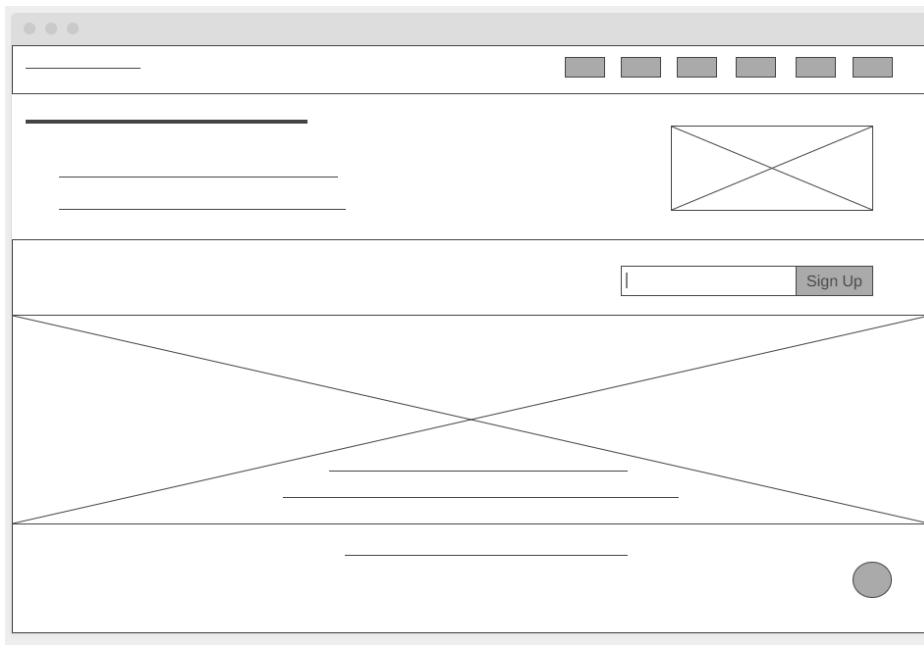
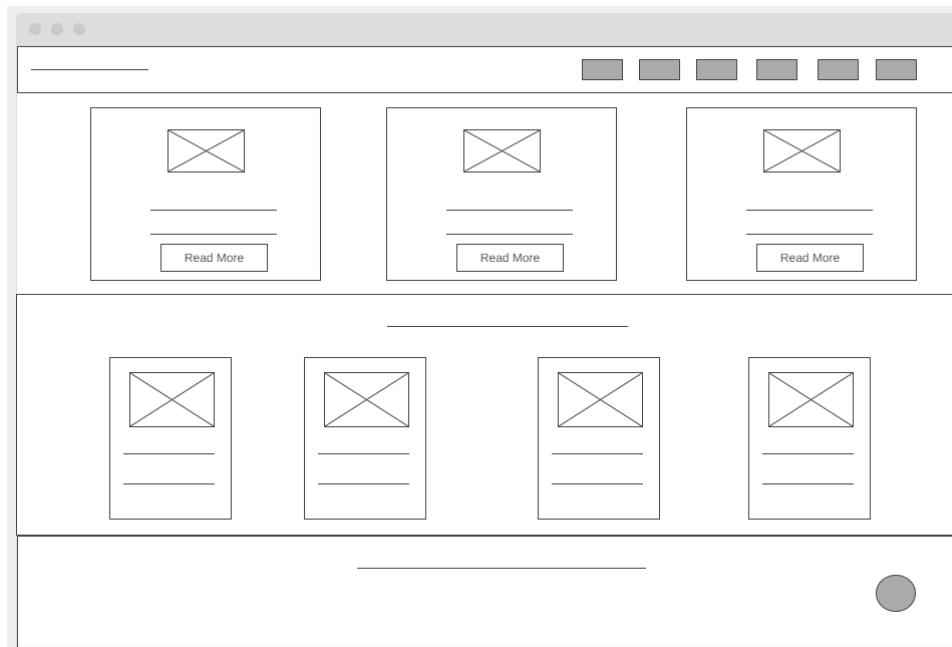


Figure 3. 197 Home Page Difference between multipage website created (Final Output) to the Designed Document(Wireframe)

About Us Page

Table 3. 6 About Us Page Difference between multipage website created (Final Output) to the Designed Document(Wireframe)

| Final Output | Wireframe (Initial Design) |
|---------------------------|---|
| Contains text. | No text (Lines represent text). |
| Contains actual pictures. | No actual picture (Box with cross represent picture). |
| Buttons are functional. | Buttons are non-functional (Only representing buttons). |
| Rich with colors. | No colors. |
| Has logo. | Footer has no logo. |



localhost/Arogya/Page_Files/Details_Pages/About.php

AROGYA Home About Us Blog Contact Us Services ▾ Supplies ▾

Awarded
We have been recognized for our exceptional service and care by both international and local awarding bodies.
[Read More>](#)

Secured
Your health is our priority. We give our priority to helping you, healing you, and protecting you.
[Read More>](#)

Excellent Staff
We have an expert medical staff to provide the best health recovery possible. So don't worry about your health.
[Read More>](#)

Our Specialists

Our Specialists all have remarkable working experience with international recognition.



Prof. Shaman



Prof. Jithangi

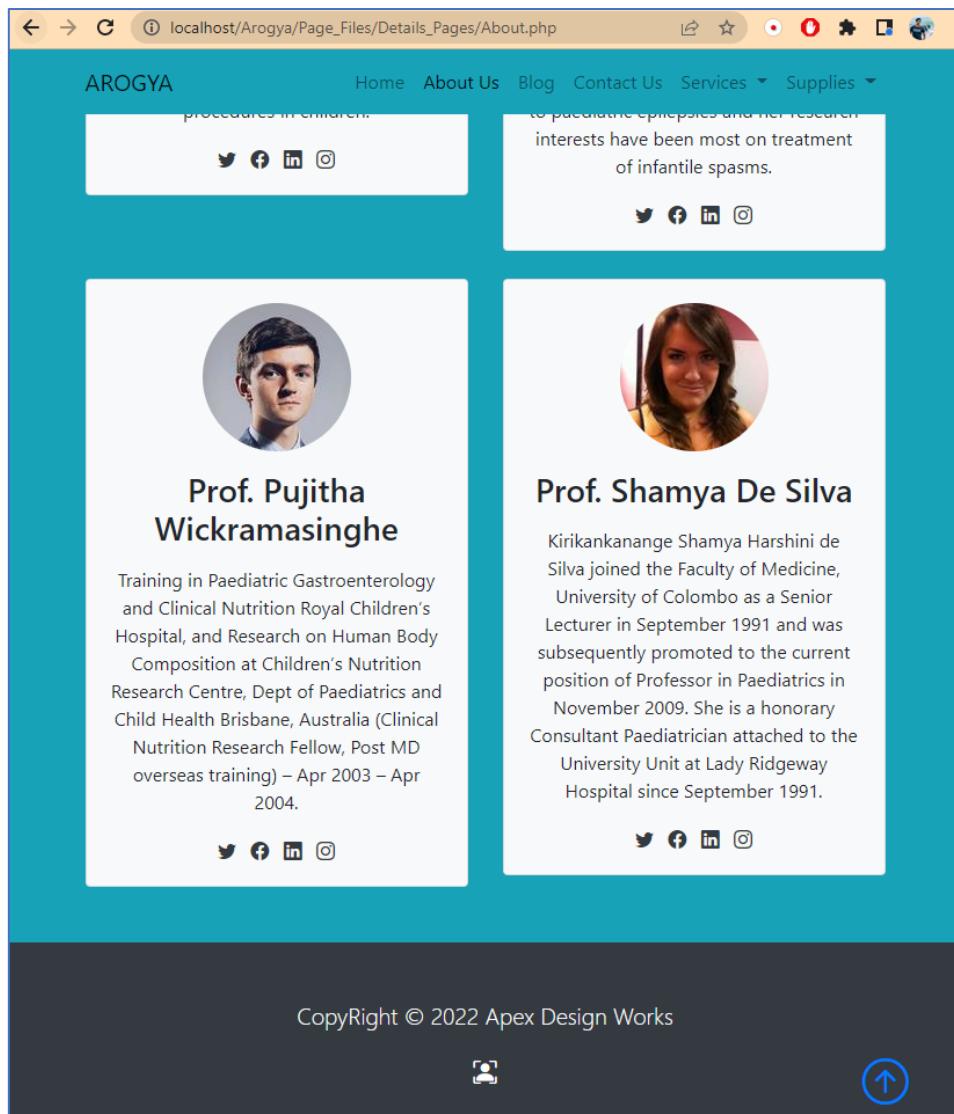
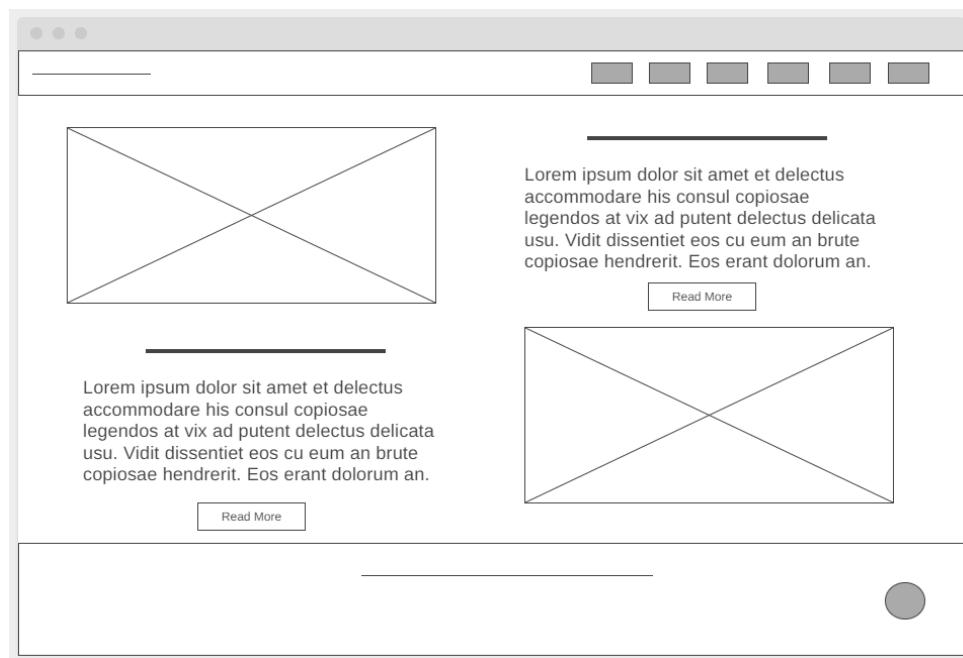


Figure 3. 198 About Us Page Difference between multipage website created (Final Output) to the Designed Document(Wireframe)

Blog Page

Table 3. 7 Blog Page Difference between multipage website created (Final Output) to the Designed Document(Wireframe)

| Final Output | Wireframe (Initial Design) |
|---------------------------|---|
| Contains text. | Contains dummy text. |
| Contains actual pictures. | No actual picture (Box with cross represent picture). |
| Buttons are functional. | Buttons are non-functional (Only representing buttons). |
| Rich with colors. | No colors. |
| Has logo. | Footer has no logo. |



A screenshot of a website titled "AROGYA". The navigation bar includes links for Home, About Us, Blog, Contact Us, Services, and Supplies. The main content area features a large title "COVID-19 Treatment". Below the title is a paragraph of text: "We offer level 1 care in four luxury hotels in the vicinity of Lanka Hospitals for asymptomatic & mildly symptomatic patients, with 24/7 hour medical & nursing care under the supervision of specialised consultants with advanced High Dependency Care." To the left of the text is a photograph of a doctor in a white coat and mask examining a patient's ear with a stethoscope. To the right of the text is another paragraph: "We provide level 2 care in a purpose-built COVID-19 ICU with critical care beds dedicated for patients who need ICU care with the best critical care doctors & nurses, ventilator support, 24-hour monitoring, high-flow oxygen, all investigations and medication." A "Read More" button is located at the bottom of this section.

AROGYA

Home About Us Blog Contact Us Services Supplies

Patient Care Centre

International Patient Care Centre is a dedicated centre at Lanka Hospitals designed to provide personalized, quality care for international patients travelling to Sri Lanka from overseas. The centre offers a plethora of services ranging from attentive patient care and monitoring to travel and accommodation arrangements, liaising with relevant parties for visa arrangements and much more. We ensure that you are well looked after by our team of trained health professionals.

> Read More

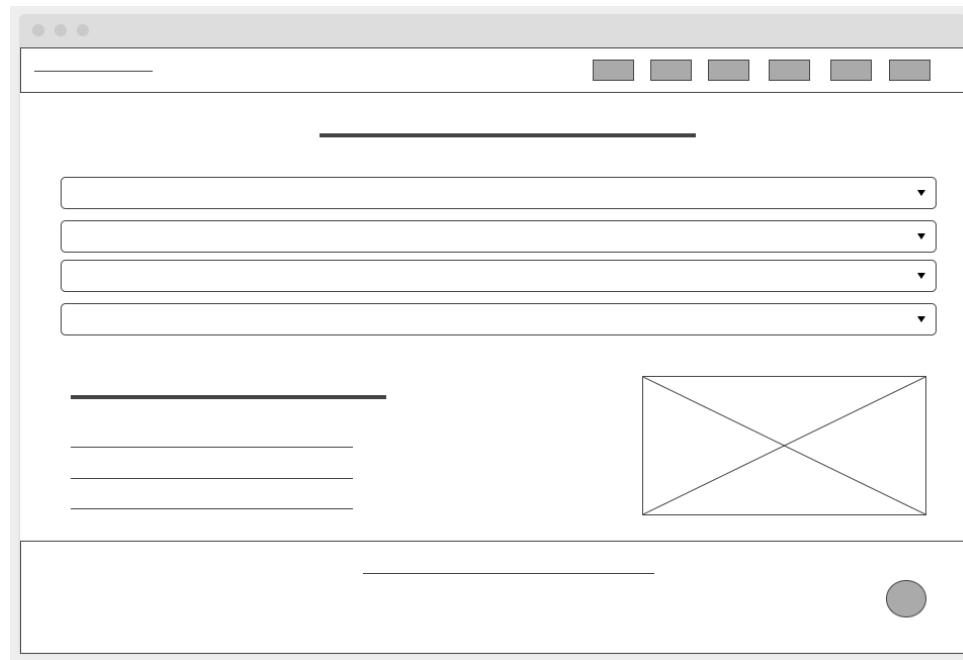
CopyRight © 2022 Apex Design Works

Figure 3. 199 Blog Page Difference between multipage website created (Final Output) to the Designed Document(Wireframe)

Contact Us Page

Table 3. 8 Contact Us Page Difference between multipage website created (Final Output) to the Designed Document(Wireframe)

| Final Output | Wireframe (Initial Design) |
|---|---|
| Contains text. | No text (Lines represent text). |
| Contains actual pictures. | No actual picture (Box with cross represent picture). |
| Buttons are functional. | Buttons are non-functional (Only representing buttons). |
| Rich with colors. | No colors. |
| Has logo. | Footer has no logo. |
| All 4 accordions are functional. | Only representing accordion. |
| Mapbox is functional (Can Zoom in and Zoom out) | Only representing a picture. |



localhost/Arogya/Page_Files/Details_Pages/Contact.php

AROGYA Home About Us Blog Contact Us Services ▾ Supplies ▾

Frequently Asked Questions

What procedures will I have in the Emergency Department? ^

Treatments and procedures may include lab work, radiological studies, CAT scans, ultrasound, EKG for cardiac review, medication administration and physician evaluations.

How is the Emergency Department staffed? ▼

Will I be allowed to have visitors during my treatment? ▼

What happens if I am admitted? ▼

Contact Info

Main Location: 578 Elvitigala Mawatha, Colombo 00500

Phone: +94 11 754 4801

Office Email: INFO@AROGYA.LK



Mapbox © OpenStreetMap Improve this map

CopyRight © 2022 Apex Design Works

Figure 3. 200 Contact Us Page Difference between multipage website created (Final Output) to the Designed Document(Wireframe)

Staff Page

Table 3. 9 Staff Page Difference between multipage website created (Final Output) to the Designed Document(Wireframe)

| Final Output | Wireframe (Initial Design) |
|---|---|
| Contains text. | No text (Lines represent text). |
| Contains icons to enhance the quality of button. | No icons. |
| Buttons are functional. | Buttons are non-functional (Only representing buttons). |
| Rich with colors. | No colors. |
| Some buttons (Create, View, and Edit) are exclusively made for redirecting to another page. | Only dummy buttons. |

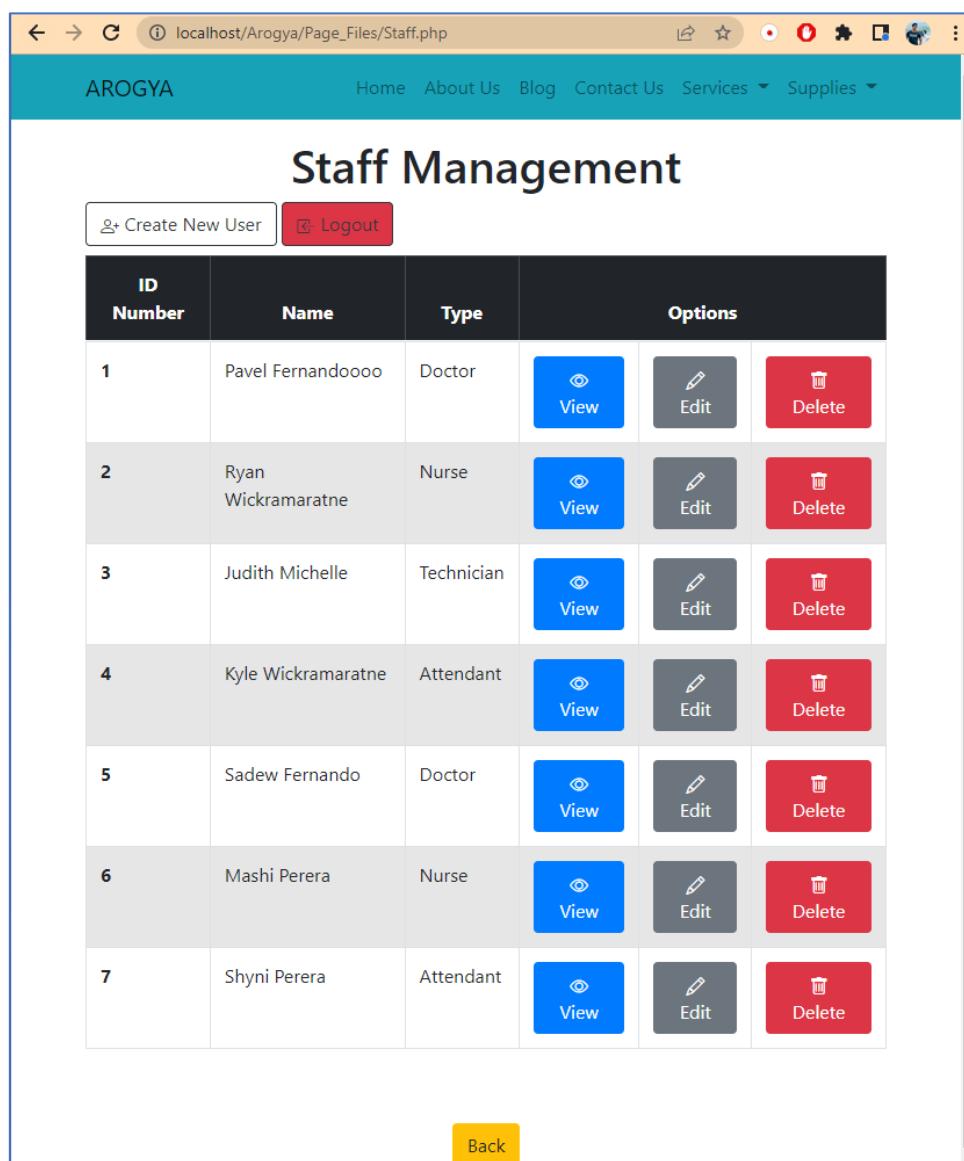
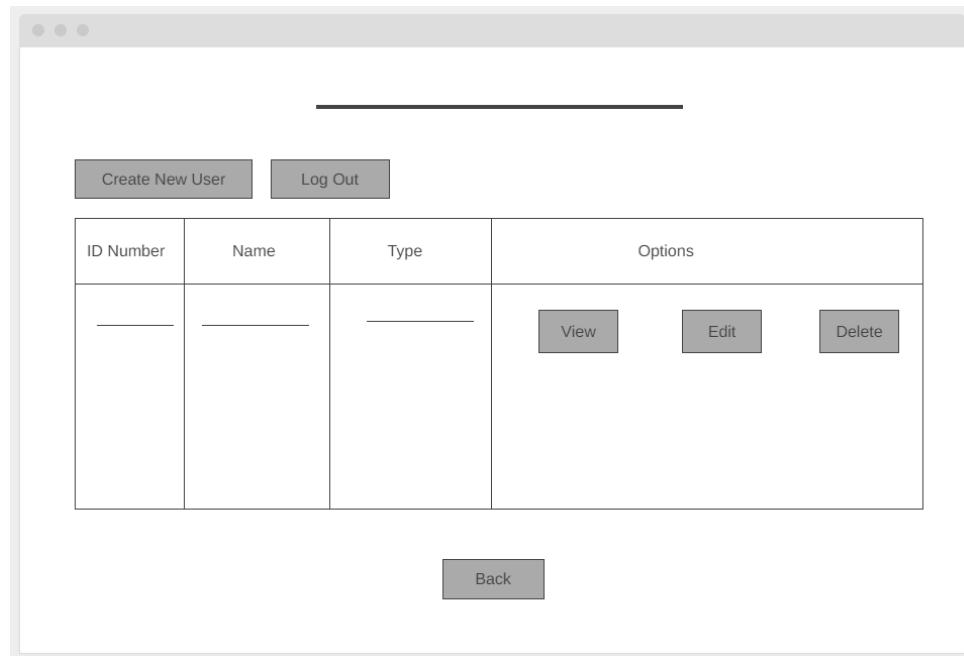
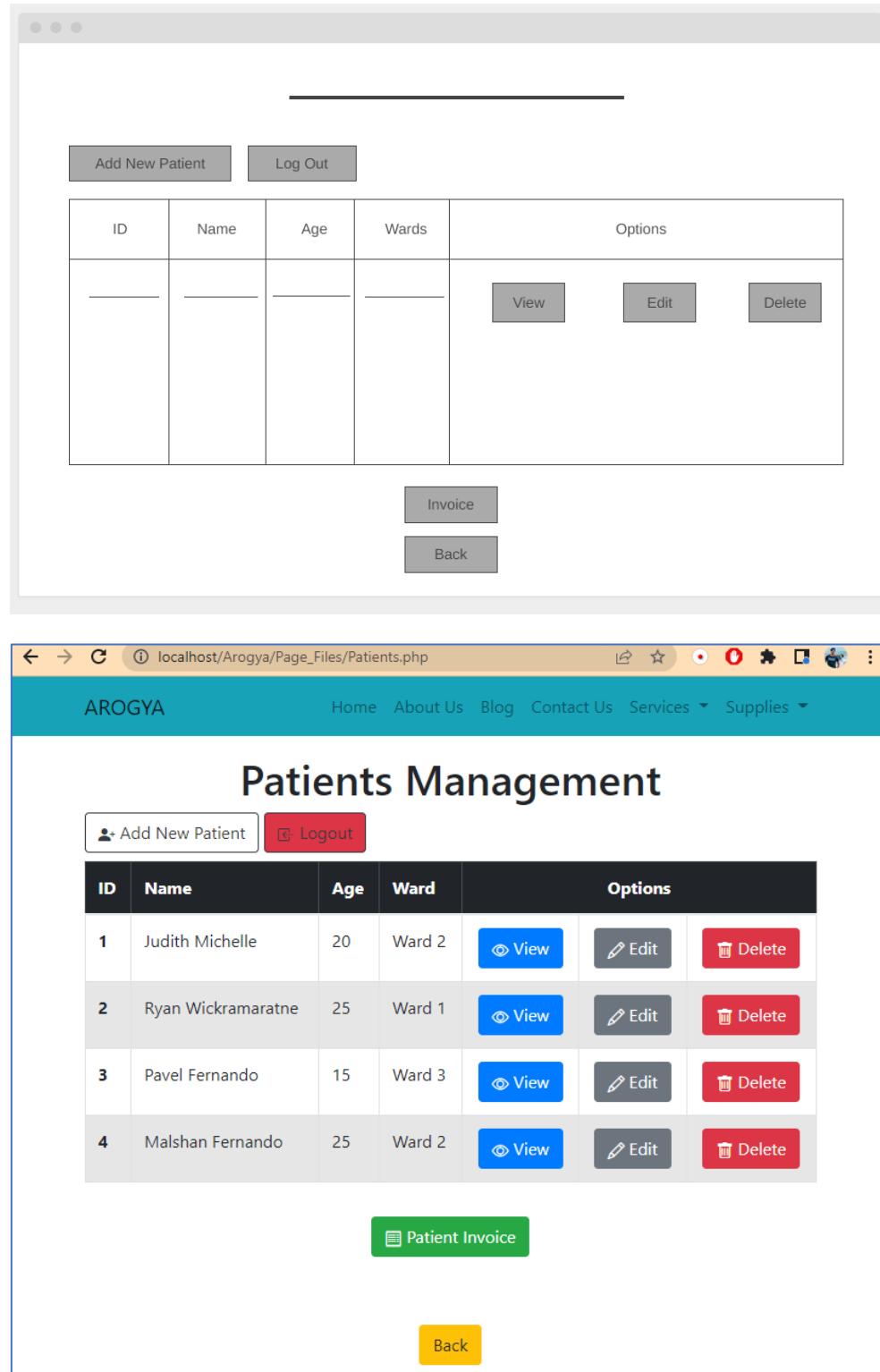


Figure 3. 201 Staff Page Difference between multipage website created (Final Output) to the Designed Document(Wireframe)

Patients Page

Table 3. 10 Patients Page Difference between multipage website created (Final Output) to the Designed Document(Wireframe)

| Final Output | Wireframe (Initial Design) |
|---|---|
| Contains text. | No text (Lines represent text). |
| Contains icons to enhance the quality of button. | No icons. |
| Buttons are functional. | Buttons are non-functional (Only representing buttons). |
| Rich with colors. | No colors. |
| Some buttons (Create, View, and Edit) are exclusively made for redirecting to another page. | Only dummy buttons. |
| Patient invoice button is exclusively made for redirecting to modal. | Only dummy button. |



The image shows a comparison between a wireframe and a final output for a patients management page. The top part is a wireframe with a light gray background, showing a header with 'Add New Patient' and 'Log Out' buttons, a table with columns 'ID', 'Name', 'Age', 'Wards', and 'Options' (containing 'View', 'Edit', and 'Delete' buttons), and footer buttons for 'Invoice' and 'Back'. The bottom part is a screenshot of a web browser displaying the final output. The browser's address bar shows 'localhost/Arogya/Page_Files/Patients.php'. The page has a teal header with the 'AROGYA' logo and navigation links: Home, About Us, Blog, Contact Us, Services, and Supplies. Below the header is a title 'Patients Management'. The main content area contains a table with the same structure as the wireframe, listing four patient entries with ID numbers 1, 2, 3, and 4, names like Judith Michelle, Ryan Wickramaratne, Pavel Fernando, and Malshan Fernando, ages, wards, and 'View', 'Edit', and 'Delete' buttons. At the bottom are a green 'Patient Invoice' button and a yellow 'Back' button.

Figure 3. 202 Patients Page Difference between multipage website created (Final Output) to the Designed Document(Wireframe)

Wards Page

Table 3. 11 Wards Page Difference between multipage website created (Final Output) to the Designed Document(Wireframe)

| Final Output | Wireframe (Initial Design) |
|--|---|
| Contains text. | No text (Lines represent text). |
| Contains icons to enhance the quality of button. | No icons. |
| Buttons are functional. | Buttons are non-functional (Only representing buttons). |
| Rich with colors. | No colors. |
| Some buttons (Manage the ward) are exclusively made for redirecting to another page. | Only dummy buttons. |

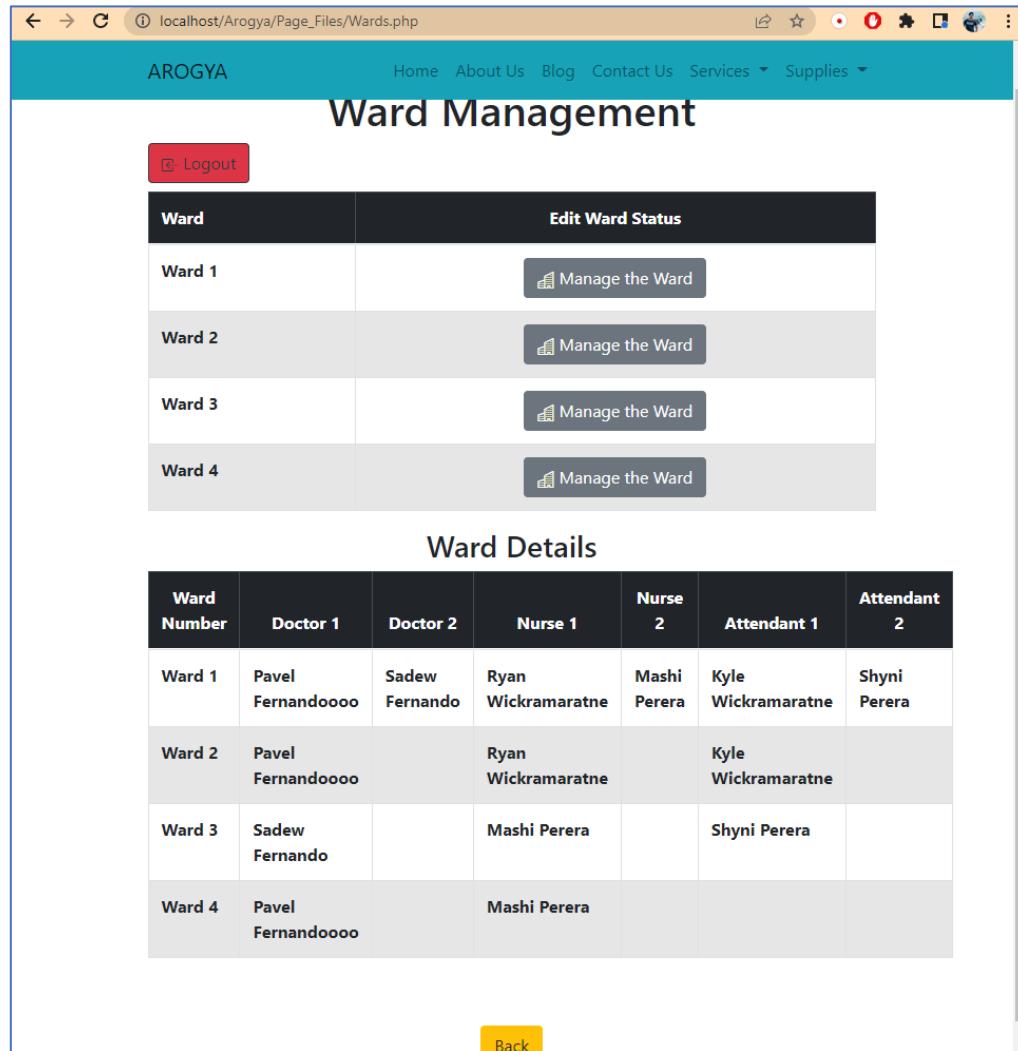
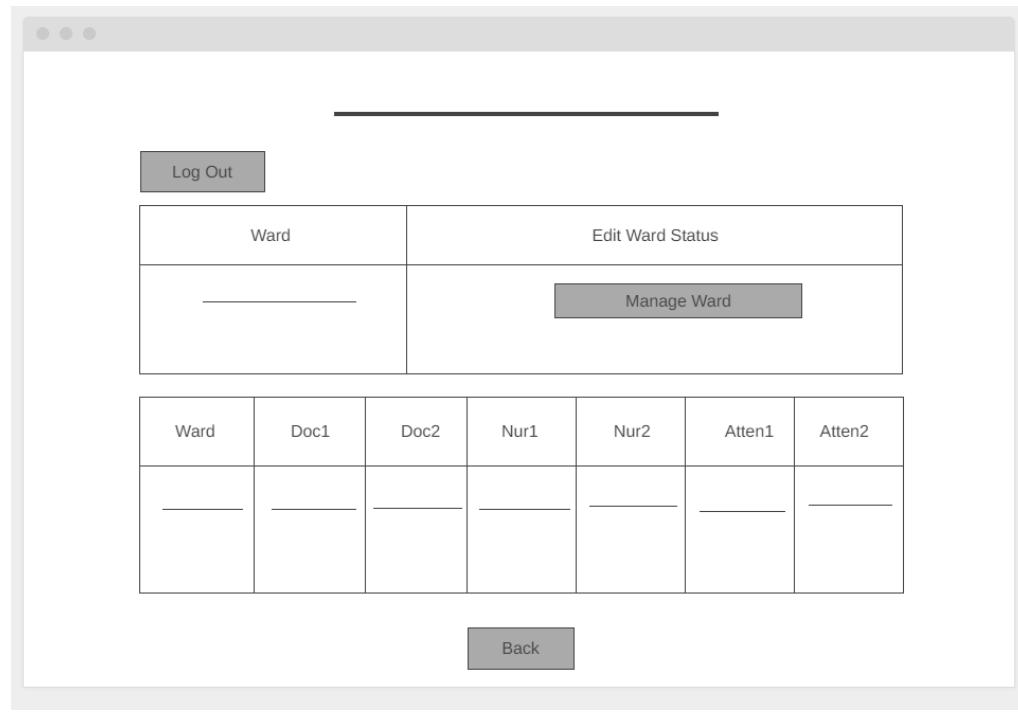


Figure 3. 203 Wards Page Difference between multipage website created (Final Output) to the Designed Document(Wireframe)

Rooms Page

Table 3. 12 Rooms Page Difference between multipage website created (Final Output) to the Designed Document(Wireframe)

| Final Output | Wireframe (Initial Design) |
|--|---|
| Contains text. | No text (Lines represent text). |
| Contains icons to enhance the quality of button. | No icons. |
| Buttons are functional. | Buttons are non-functional (Only representing buttons). |
| Rich with colors. | No colors. |
| Some buttons (Manage the ward rooms) are exclusively made for redirecting to another page. | Only dummy buttons. |

The image displays two versions of a 'Room Management' page. The top part is a wireframe, and the bottom part is the final website output.

Wireframe (Top):

- A header bar with three dots and a horizontal line.
- A 'Log Out' button.
- A table with two rows:
 - Row 1: 'Ward' and 'Edit Ward RoomStatus' button.
 - Row 2: A blank row and a 'Manage Rooms' button.

Final Output (Bottom):

- A browser header with back, forward, and search icons, and the URL 'localhost/Arogya/Page_Files/Rooms.php'.
- A navigation bar with 'AROGYA' and links: Home, About Us, Blog, Contact Us, Services, and Supplies.
- A main title 'Room Management'.
- A 'Logout' button.
- A table titled 'Edit Ward Room Status' with four rows:
 - Row 1: 'Ward' column and 'Manage the Ward Rooms' button.
 - Row 2: 'Ward 1' and 'Manage the Ward Rooms' button.
 - Row 3: 'Ward 2' and 'Manage the Ward Rooms' button.
 - Row 4: 'Ward 3' and 'Manage the Ward Rooms' button.
- A section titled 'Ward Rooms Details' with a table:

| Ward Number | Room 1 | Room 2 | Room 3 | Room 4 |
|-------------|-----------|-----------|-----------|-----------|
| Ward 1 | No | No | Available | Available |
| Ward 2 | Available | Available | Available | Available |
| Ward 3 | No | No | No | Available |
| Ward 4 | No | No | No | No |
- A 'Back' button.

Figure 3. 204 Rooms Page Difference between multipage website created (Final Output) to the Designed Document(Wireframe)

Medicines Page

Table 3. 13 Medicines Page Difference between multipage website created (Final Output) to the Designed Document(Wireframe)

| Final Output | Wireframe (Initial Design) |
|---|---|
| Contains text. | No text (Lines represent text). |
| Contains icons to enhance the quality of button. | No icons. |
| Buttons are functional. | Buttons are non-functional (Only representing buttons). |
| Rich with colors. | No colors. |
| Some buttons (Create, View, and Edit) are exclusively made for redirecting to another page. | Only dummy buttons. |

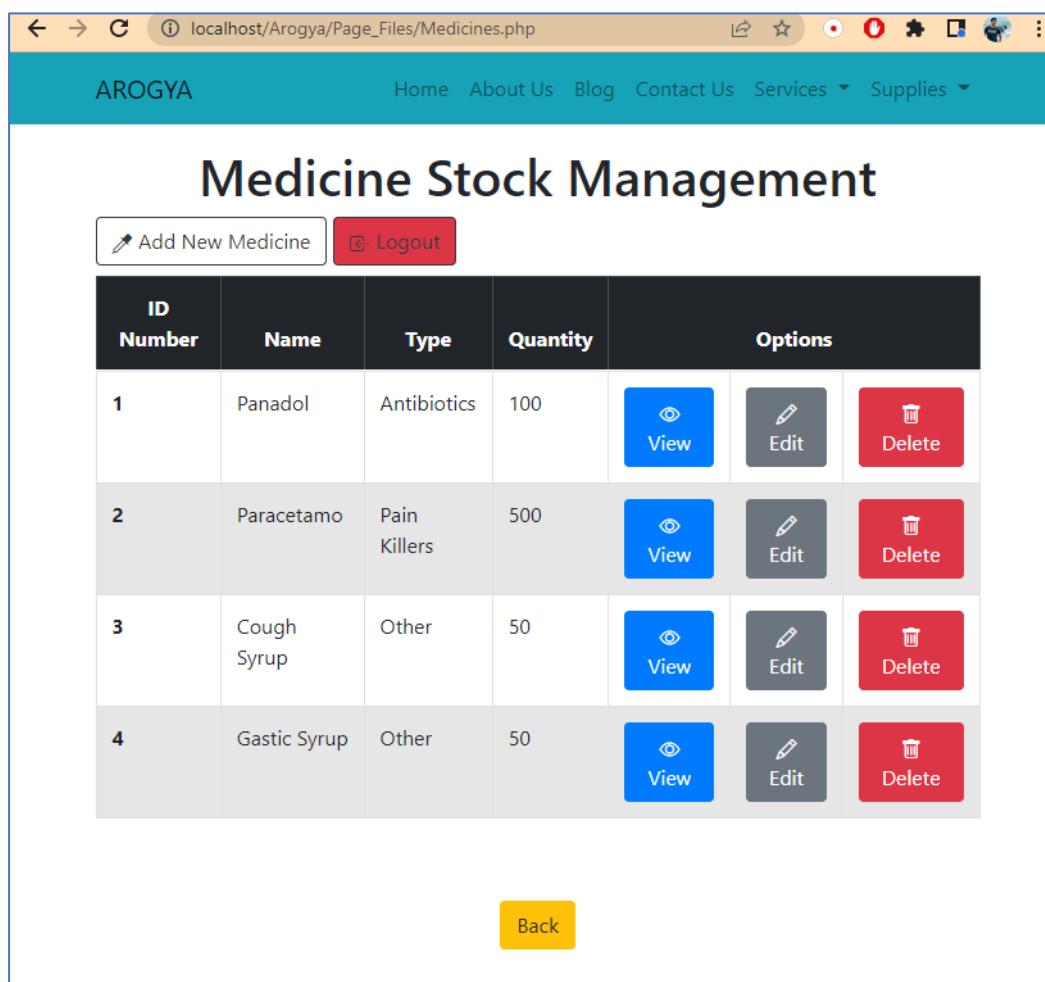
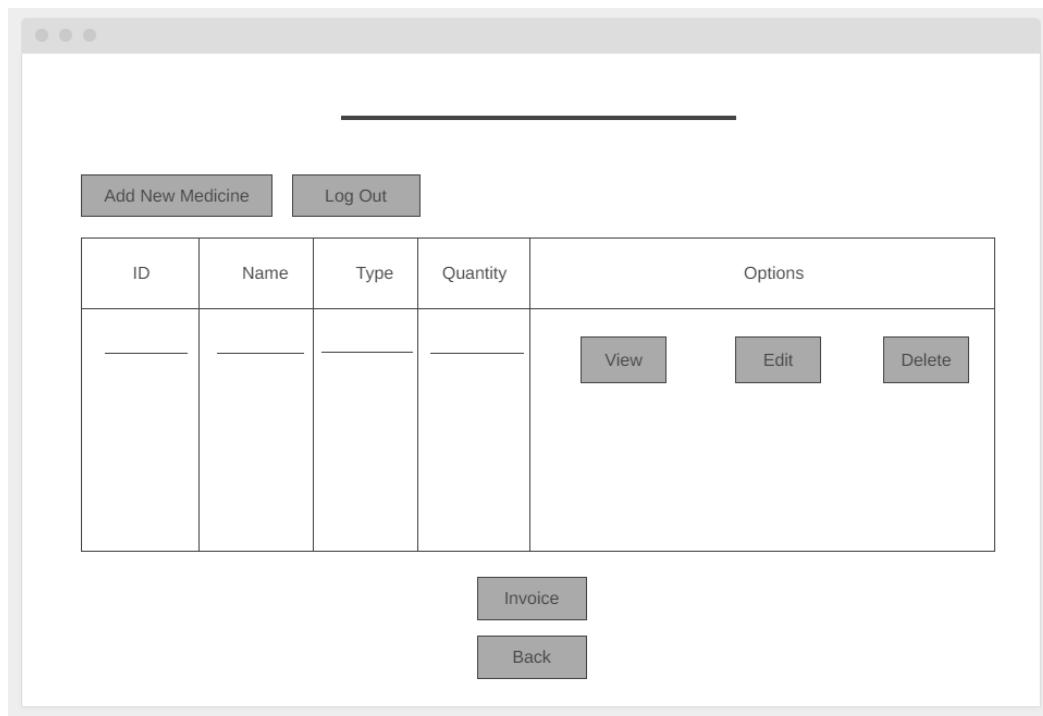


Figure 3. 205 Medicines Page Difference between multipage website created (Final Output) to the Designed Document(Wireframe)

Other Page

Table 3. 14 Other Page Difference between multipage website created (Final Output) to the Designed Document(Wireframe)

| Final Output | Wireframe (Initial Design) |
|---|---|
| Contains text. | No text (Lines represent text). |
| Contains icons to enhance the quality of button. | No icons. |
| Buttons are functional. | Buttons are non-functional (Only representing buttons). |
| Rich with colors. | No colors. |
| Some buttons (Create, View, and Edit) are exclusively made for redirecting to another page. | Only dummy buttons. |

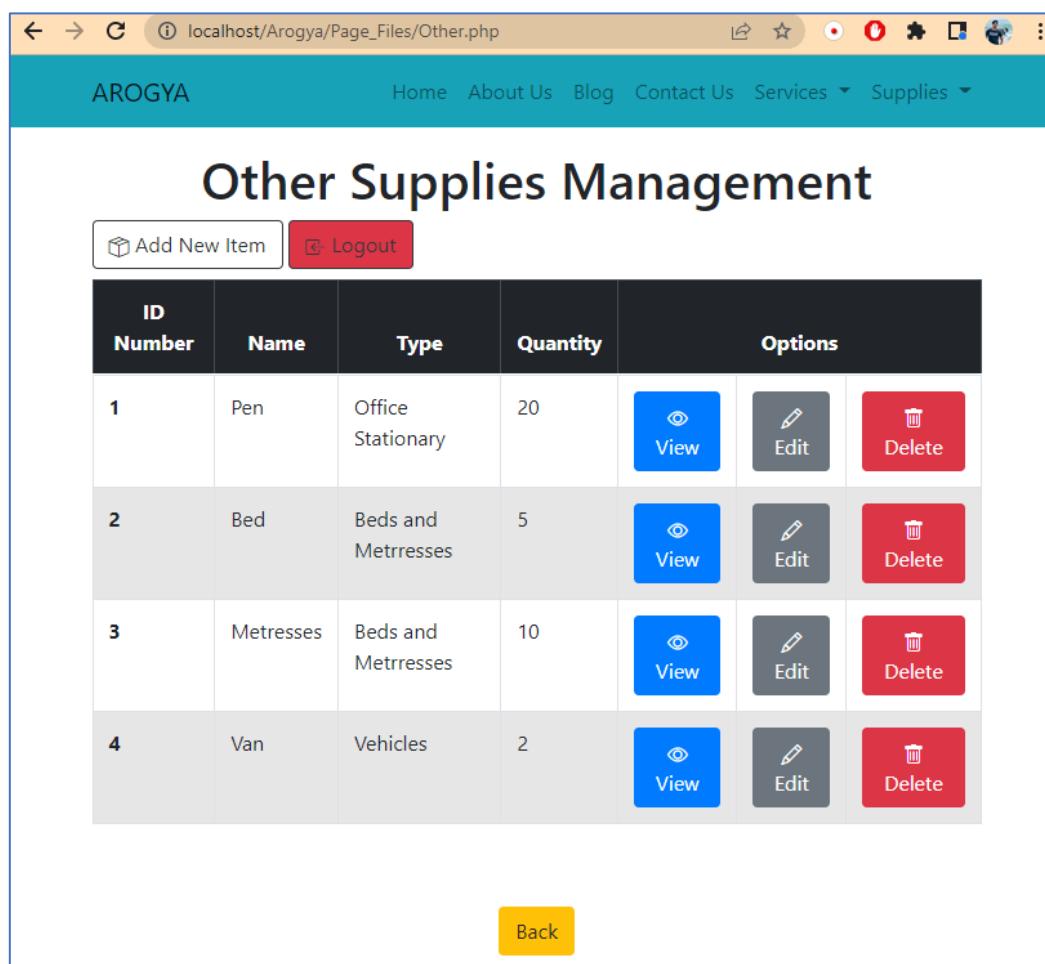
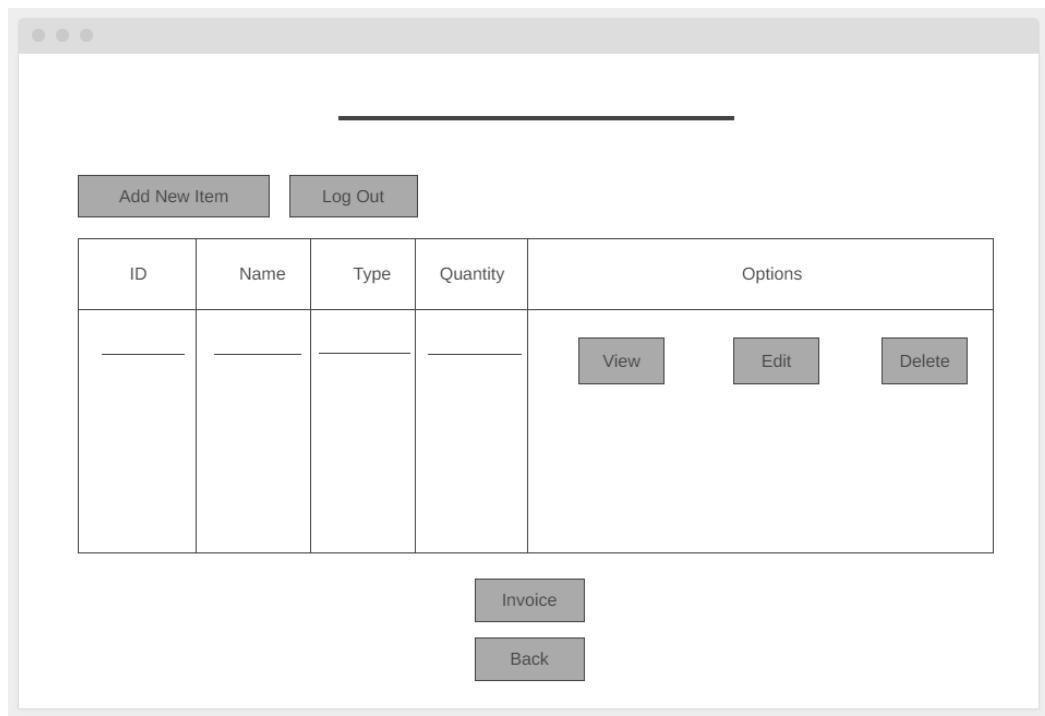


Figure 3. 206 Other Page Difference between multipage website created (Final Output) to the Designed Document(Wireframe)

Invoice Modal

Table 3. 15 Invoice Modal Difference between multipage website created (Final Output) to the Designed Document(Wireframe)

| Final Output | Wireframe (Initial Design) |
|---|---|
| Contains text. | No text (Lines represent text). |
| Contains icons to enhance the quality. | No icons. |
| Buttons are functional. | Buttons are non-functional (Only representing buttons). |
| Rich with colors. | No colors. |
| “Check me out” check box available. | No check boxes. |
| Top close button available. | No top close button. |
| When opening and closing the modal, it's fade away. | Can't represent functions. |

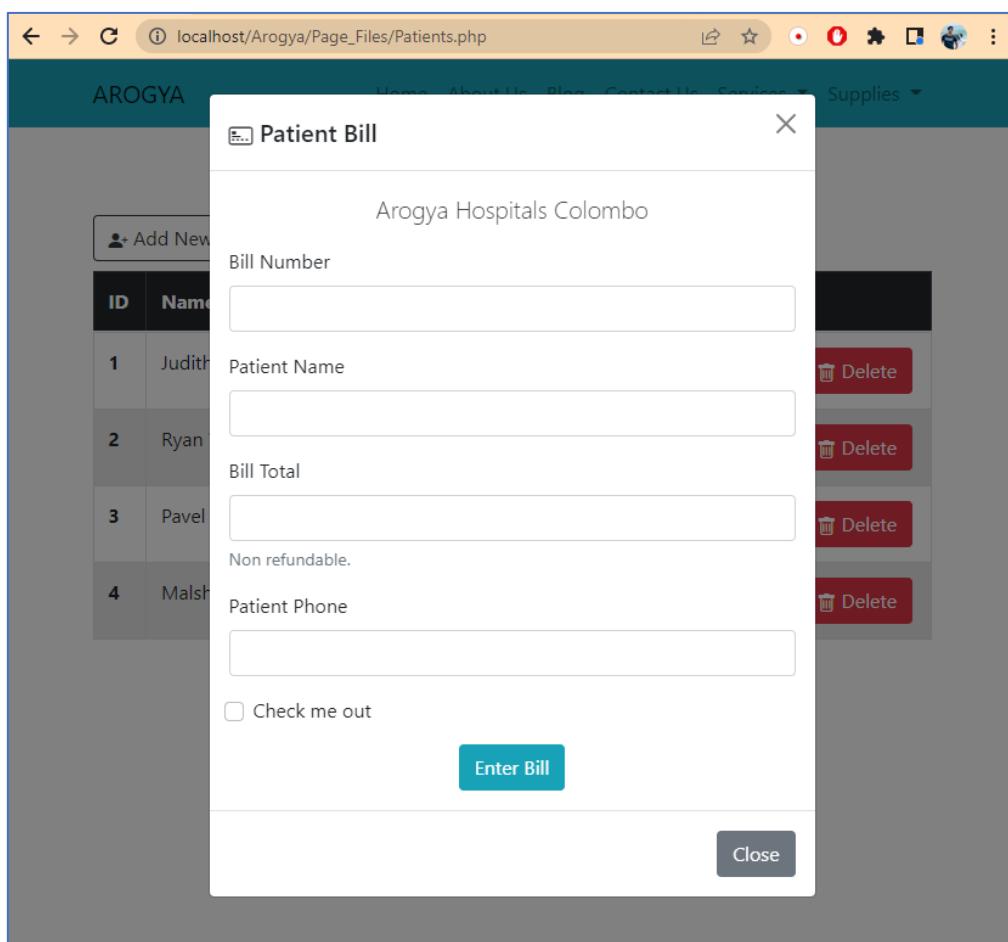
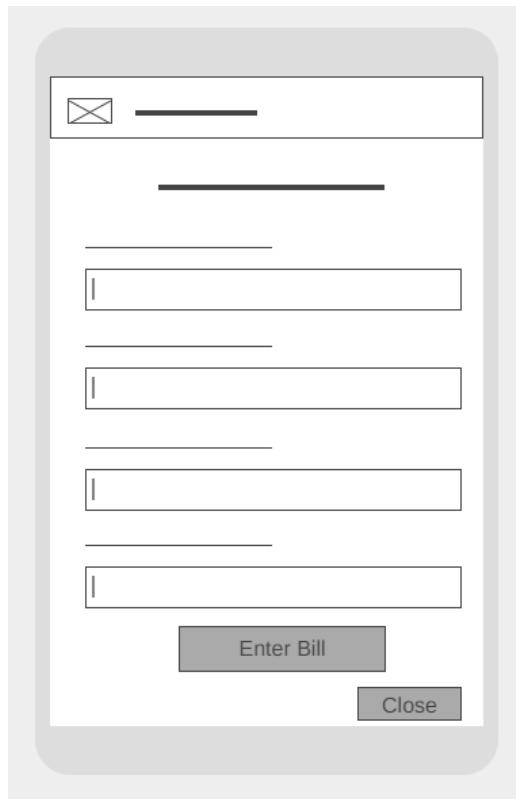


Figure 3. 207 Invoice Modal Difference between multipage website created (Final Output) to the Designed Document(Wireframe)

3.4.3 Compare and Contrast each final output's pages with the initial output

Login Page

Table 3. 16 Login Page Difference between multipage website created (Final Output) to the initial output

| Final Output | Initial Output |
|------------------------|-----------------------|
| Contains text. | Contains text. |
| Contains pictures. | No pictures. |
| Button is functional. | Button is functional. |
| Rich with colors. | No colors. |
| Placeholder available. | No placeholder. |

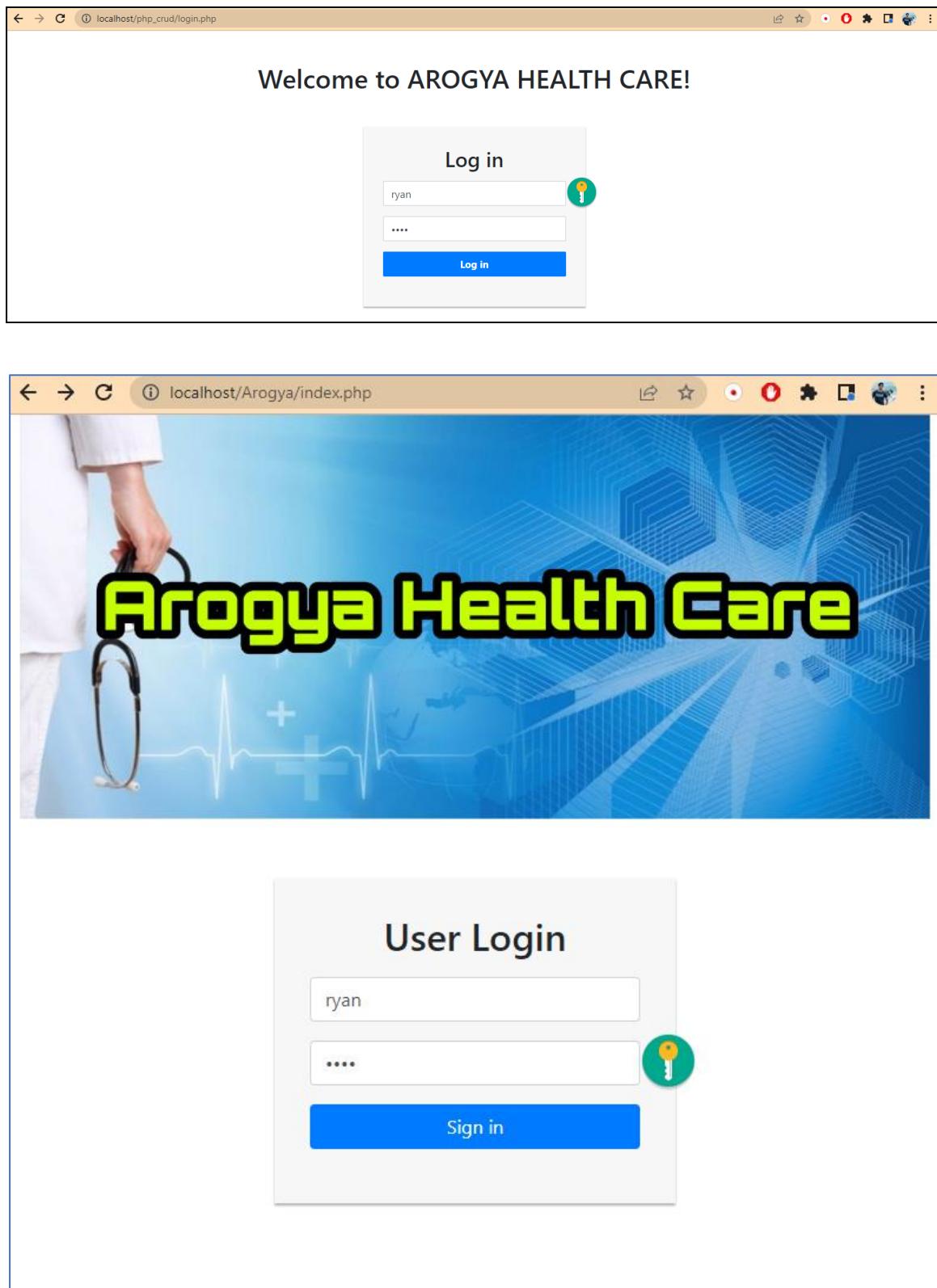


Figure 3. 208 Login Page Difference between multipage website created (Final Output) to the initial output

Home Page

Table 3. 17 Home Page Difference between multipage website created (Final Output) to the initial output

| Final Output | Initial Output |
|---------------------------------|--|
| Contains text. | |
| Contains actual pictures. | |
| Only 1 picture is representing. | Didn't design this page since it's not critical requirement. |
| Buttons are functional. | |
| Rich with colors. | |
| Has logo in the footer. | |

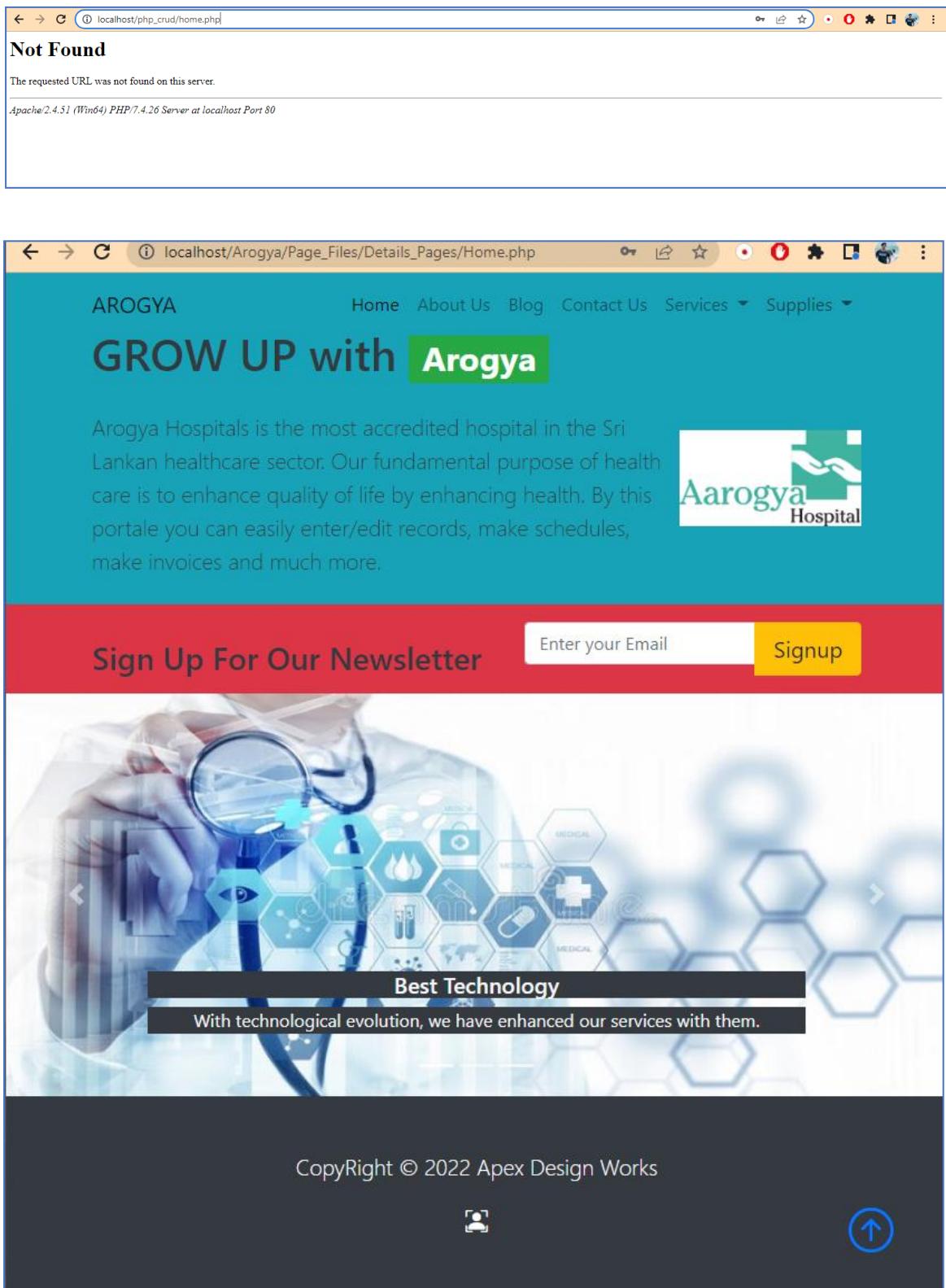


Figure 3. 209 Home Page Difference between multipage website created (Final Output) to the initial output

About Us Page

Table 3. 18 About Us Page Difference between multipage website created (Final Output) to the initial output

| Final Output | Initial Output |
|---------------------------|--|
| Contains text. | |
| Contains actual pictures. | |
| Buttons are functional. | Didn't design this page since it's not critical requirement. |
| Rich with colors. | |
| Has logo in the footer. | |



AROGYA

Home About Us Blog Contact Us Services ▾ Supplies ▾

Awarded

We have been recognized for our exceptional service and care by both international and local awarding bodies.

[Read More>](#)

Secured

Your health is our priority. We give our priority to helping you, healing you, and protecting you.

[Read More>](#)

Excellent Staff

We have an expert medical staff to provide the best health recovery possible. So don't worry about your health.

[Read More>](#)

Our Specialists

Our Specialists all have remarkable working experience with international recognition.

Prof. Shaman

Prof. Jithangi

AROGYA

procedures in children.

Home About Us Blog Contact Us Services ▾ Supplies ▾

to paediatric Epilepsies and her research interests have been most on treatment of infantile spasms.

Prof. Pujitha Wickramasinghe

Training in Paediatric Gastroenterology and Clinical Nutrition Royal Children's Hospital, and Research on Human Body Composition at Children's Nutrition Research Centre, Dept of Paediatrics and Child Health Brisbane, Australia (Clinical Nutrition Research Fellow, Post MD overseas training) – Apr 2003 – Apr 2004.

Prof. Shamya De Silva

Kirikankanage Shamya Harshini de Silva joined the Faculty of Medicine, University of Colombo as a Senior Lecturer in September 1991 and was subsequently promoted to the current position of Professor in Paediatrics in November 2009. She is a honorary Consultant Paediatrician attached to the University Unit at Lady Ridgeway Hospital since September 1991.

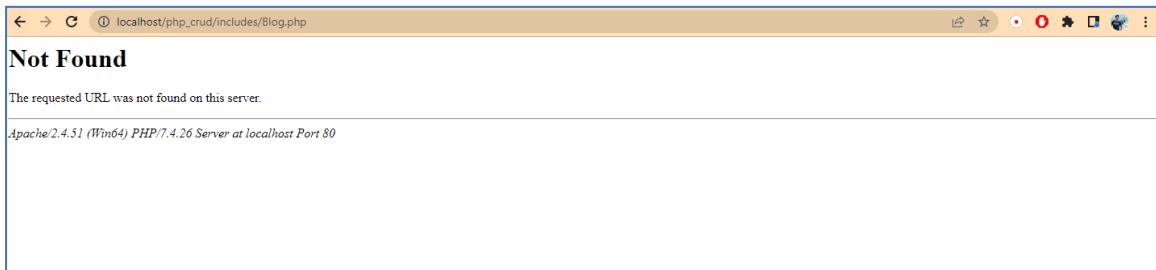
CopyRight © 2022 Apex Design Works

Figure 3. 210 About Us Page Difference between multipage website created (Final Output) to the initial output

Blog Page

Table 3. 19 Blog Page Difference between multipage website created (Final Output) to the initial output

| Final Output | Initial Output |
|---------------------------|--|
| Contains text. | |
| Contains actual pictures. | |
| Buttons are functional. | Didn't design this page since it's not critical requirement. |
| Rich with colors. | |
| Has logo in the footer. | |



The screenshot shows a web browser window with the URL localhost/Arogya/Page_Files/Details_Pages/Blog.php. The page has a teal header bar with the logo 'AROGYA' and navigation links: Home, About Us, Blog, Contact Us, Services ▾, and Supplies ▾. Below the header is a large image of a doctor in a white coat examining a patient's arm. To the right of the image is a section titled 'COVID-19 Treatment' with two columns of text describing medical care levels and services.

COVID-19 Treatment

We offer level 1 care in four luxury hotels in the vicinity of Lanka Hospitals for asymptomatic & mildly symptomatic patients, with 24/7 hour medical & nursing care under the supervision of specialised consultants with advanced High Dependency Care.

We provide level 2 care in a purpose-built COVID-19 ICU with critical care beds dedicated for patients who need ICU care with the best critical care doctors & nurses, ventilator support, 24-hour monitoring, high-flow oxygen, all investigations and medication.

> [Read More](#)

The screenshot shows a web browser window with the URL `localhost/Arogya/Page_Files/Details_Pages/Blog.php`. The page title is "Patient Care Centre". The main content area contains a paragraph describing the International Patient Care Centre as a dedicated centre at Lanka Hospitals designed to provide personalized, quality care for international patients travelling to Sri Lanka from overseas. It mentions services ranging from attentive patient care and monitoring to travel and accommodation arrangements, liaising with relevant parties for visa arrangements, and much more. The text is attributed to a team of trained health professionals. A "Read More" link is visible at the bottom of the text block. To the right of the text, there is a photograph of a hospital ward with multiple beds and medical equipment. The footer of the page includes copyright information for Apex Design Works and icons for user profile and upload.

AROGYA

Home About Us Blog Contact Us Services ▾ Supplies ▾

Patient Care Centre

International Patient Care Centre is a dedicated centre at Lanka Hospitals designed to provide personalized, quality care for international patients travelling to Sri Lanka from overseas. The centre offers a plethora of services ranging from attentive patient care and monitoring to travel and accommodation arrangements, liaising with relevant parties for visa arrangements and much more. We ensure that you are well looked after by our team of trained health professionals.

> Read More

CopyRight © 2022 Apex Design Works

Figure 3. 211 Blog Page Difference between multipage website created (Final Output) to the initial output

Contact Us Page

Table 3. 20 Contact Us Page Difference between multipage website created (Final Output) to the initial output

| Final Output | Initial Output |
|---|--|
| Contains text. | |
| Contains actual pictures. | |
| Buttons are functional. | |
| Rich with colors. | Didn't design this page since it's not critical requirement. |
| Has logo in the footer. | |
| All 4 accordions are functional. | |
| Mapbox is functional (Can Zoom in and Zoom out) | |



localhost/Arogya/Page_Files/Details_Pages/Contact.php

AROGYA Home About Us Blog Contact Us Services ▾ Supplies ▾

Frequently Asked Questions

What procedures will I have in the Emergency Department? ^

Treatments and procedures may include lab work, radiological studies, CAT scans, ultrasound, EKG for cardiac review, medication administration and physician evaluations.

How is the Emergency Department staffed? ▼

Will I be allowed to have visitors during my treatment? ▼

What happens if I am admitted? ▼

Contact Info

Main Location: 578 Elvitigala Mawatha, Colombo 00500

Phone: +94 11 754 4801

Office Email: INFO@AROGYA.LK

© Mapbox © OpenStreetMap Improve this map

CopyRight © 2022 Apex Design Works

Figure 3. 212 Contact Us Page Difference between multipage website created (Final Output) to the initial output

Staff Page

Table 3. 21 Staff Page Difference between multipage website created (Final Output) to the initial output

| Final Output | Initial Output |
|---|--|
| Contains text. | Contains text. |
| Contains icons to enhance the quality of button. | Contains icons to enhance the quality of button. |
| Buttons are functional. | Buttons are functional. |
| Rich with colors. | Reduced colors. |
| Some buttons (Create, View, and Edit) are exclusively made for redirecting to another page. | Only dummy buttons. |

localhost/php_crud/includes/home.php

AROGYA HEALTH CARE Home Patients Schedule

STAFF MANAGEMENT

[Create New User](#) [Logout](#)

| ID | Name | Type | CRUD Operations | | |
|----|--------------------|------------|----------------------|----------------------|------------------------|
| 1 | Pavel Fernandoooo | Doctor | View | Edit | Delete |
| 2 | Ryan Wickramaratne | Nurse | View | Edit | Delete |
| 3 | Judith Michelle | Technician | View | Edit | Delete |
| 4 | Kyle Wickramaratne | Attendant | View | Edit | Delete |
| 5 | Sadew Fernando | Doctor | View | Edit | Delete |
| 6 | Mashi Perera | Nurse | View | Edit | Delete |
| 7 | Shyni Perera | Attendant | View | Edit | Delete |

[Back](#)

localhost/Arogya/Page_Files/Staff.php

AROGYA Home About Us Blog Contact Us Services Supplies

Staff Management

[Create New User](#) [Logout](#)

| ID Number | Name | Type | Options | | |
|-----------|--------------------|------------|----------------------|----------------------|------------------------|
| 1 | Pavel Fernandoooo | Doctor | View | Edit | Delete |
| 2 | Ryan Wickramaratne | Nurse | View | Edit | Delete |
| 3 | Judith Michelle | Technician | View | Edit | Delete |
| 4 | Kyle Wickramaratne | Attendant | View | Edit | Delete |
| 5 | Sadew Fernando | Doctor | View | Edit | Delete |
| 6 | Mashi Perera | Nurse | View | Edit | Delete |
| 7 | Shyni Perera | Attendant | View | Edit | Delete |

[Back](#)

Figure 3. 213 Staff Page Difference between multipage website created (Final Output) to the initial output

Patients Page

Table 3. 22 Patients Page Difference between multipage website created (Final Output) to the initial output

| Final Output | Initial Output |
|---|--|
| Contains text. | Contains text. |
| Contains icons to enhance the quality of button. | Contains icons to enhance the quality of button. |
| Buttons are functional. | Buttons are functional. |
| Rich with colors. | Reduced colors. |
| Some buttons (Create, View, and Edit) are exclusively made for redirecting to another page. | Only dummy buttons. |
| Patient invoice button is exclusively made for redirecting to modal. | Only dummy button. |

localhost/php_crud/patients.php

AROGYA HEALTH CARE Home Patients Schedule

PATIENTS MANAGEMENT

Add New Patient Logout

| ID | Name | Age | Ward | CRUD Operations | | |
|----|--------------------|-----|--------|-----------------|--|--|
| 1 | Judith Michelle | 20 | Ward 2 | | | |
| 2 | Ryan Wickramaratne | 25 | Ward 1 | | | |
| 3 | Pavel Fernando | 15 | Ward 3 | | | |
| 4 | Malshan Fernando | 25 | Ward 2 | | | |

Back

localhost/Arogya/Page_Files/Patients.php

AROGYA Home About Us Blog Contact Us Services Supplies

Patients Management

Add New Patient Logout

| ID | Name | Age | Ward | Options | | |
|----|--------------------|-----|--------|---------|--|--|
| 1 | Judith Michelle | 20 | Ward 2 | | | |
| 2 | Ryan Wickramaratne | 25 | Ward 1 | | | |
| 3 | Pavel Fernando | 15 | Ward 3 | | | |
| 4 | Malshan Fernando | 25 | Ward 2 | | | |

Patient Invoice

Back

Figure 3. 214 Patients Page Difference between multipage website created (Final Output) to the initial output

Wards Page

Table 3. 23 Wards Page Difference between multipage website created (Final Output) to the initial output

| Final Output | Initial Output |
|--|--|
| Contains text. | Contains text. |
| Contains icons to enhance the quality of button. | Contains icons to enhance the quality of button. |
| Buttons are functional. | Buttons are functional. |
| Rich with colors. | Reduced colors. |
| Some buttons (Manage the ward) are exclusively made for redirecting to another page. | Only dummy buttons. |

localhost/php_crud/includes/wards.php

AROGYA HEALTH CARE Home Patients Schedule

WARDS

[Logout](#)

| WARD | STATUS |
|--------|---------------------------------|
| Ward 1 | Ward Management |
| Ward 2 | Ward Management |
| Ward 3 | Ward Management |
| Ward 4 | Ward Management |

[Back](#)

localhost/Arogya/Page_Files/Wards.php

AROGYA Home About Us Blog Contact Us Services Supplies

Ward Management

[Logout](#)

| Ward | Edit Ward Status |
|--------|---------------------------------|
| Ward 1 | Manage the Ward |
| Ward 2 | Manage the Ward |
| Ward 3 | Manage the Ward |
| Ward 4 | Manage the Ward |

Ward Details

| Ward Number | Doctor 1 | Doctor 2 | Nurse 1 | Nurse 2 | Attendant 1 | Attendant 2 |
|-------------|----------------|----------------|--------------------|--------------------|--------------------|--------------|
| Ward 1 | Pavel Fernando | Sadew Fernando | Ryan Wickramaratne | Mashi Perera | Kyle Wickramaratne | Shyni Perera |
| Ward 2 | Pavel Fernando | | Ryan Wickramaratne | Kyle Wickramaratne | | |
| Ward 3 | Sadew Fernando | | Mashi Perera | | Shyni Perera | |
| Ward 4 | Pavel Fernando | | Mashi Perera | | | |

[Back](#)

Figure 3. 215 Wards Page Difference between multipage website created (Final Output) to the initial output

Rooms Page

Table 3. 24 Rooms Page Difference between multipage website created (Final Output) to the initial output

| Final Output | Initial Output |
|--|---|
| Contains text. | |
| Contains icons to enhance the quality of button. | |
| Buttons are functional. | Didn't add this page into the system in Initial Output. |
| Rich with colors. | |
| Some buttons (Manage the ward rooms) are exclusively made for redirecting to another page. | |


Ward Number
 Room 1 | Room 2 | Room 3 | Room 4 || | | | | |
| --- | --- | --- | --- | --- |
| Ward 1 | No | No | Available | Available |
| Ward 2 | Available | Available | Available | Available |
| Ward 3 | No | No | No | Available |
| Ward 4 | No | No | No | No |

 A yellow 'Back' button is located at the bottom right."/>

Figure 3. 216 Rooms Page Difference between multipage website created (Final Output) to the initial output

Medicines Page

Table 3. 25 Medicines Page Difference between multipage website created (Final Output) to the initial output

| Final Output | Initial Output |
|---|---|
| Contains text. | |
| Contains icons to enhance the quality of button. | |
| Buttons are functional. | Didn't add this page into the system in Initial Output. |
| Rich with colors. | |
| Some buttons (Create, View, and Edit) are exclusively made for redirecting to another page. | |

The requested URL was not found on this server.
Apache/2.4.51 (Win64) PHP/7.4.26 Server at localhost Port 80

AROGYA

- [Home](#)
- [About Us](#)
- [Blog](#)
- [Contact Us](#)
- [Services](#) ▾
- [Supplies](#) ▾

Medicine Stock Management

[Add New Medicine](#) [Logout](#)

| ID Number | Name | Type | Quantity | Options | | |
|-----------|--------------|--------------|----------|---------|------|--------|
| 1 | Panadol | Antibiotics | 100 | View | Edit | Delete |
| 2 | Paracetamo | Pain Killers | 500 | View | Edit | Delete |
| 3 | Cough Syrup | Other | 50 | View | Edit | Delete |
| 4 | Gastic Syrup | Other | 50 | View | Edit | Delete |

[Back](#)

Figure 3. 217 Medicines Page Difference between multipage website created (Final Output) to the initial output

Other Page

Table 3. 26 Other Page Difference between multipage website created (Final Output) to the initial output

| Final Output | Initial Output |
|---|---|
| Contains text. | |
| Contains icons to enhance the quality of button. | |
| Buttons are functional. | Didn't add this page into the system in Initial Output. |
| Rich with colors. | |
| Some buttons (Create, View, and Edit) are exclusively made for redirecting to another page. | |



The screenshot shows a web application titled 'AROGYA'. The top navigation bar includes links for Home, About Us, Blog, Contact Us, Services, and Supplies. The main content area is titled 'Other Supplies Management'. It features two buttons: 'Add New Item' and 'Logout'. Below is a table with four rows of data:

| ID Number | Name | Type | Quantity | Options | | |
|-----------|-----------|--------------------|----------|---------|------|--------|
| 1 | Pen | Office Stationary | 20 | View | Edit | Delete |
| 2 | Bed | Beds and Metresses | 5 | View | Edit | Delete |
| 3 | Metresses | Beds and Metresses | 10 | View | Edit | Delete |
| 4 | Van | Vehicles | 2 | View | Edit | Delete |

A yellow 'Back' button is located at the bottom right of the content area.

Figure 3. 218 Other Page Difference between multipage website created (Final Output) to the initial output

3.4.4 Technical challenges faced during the development.

An attractive design is often the first thing that draws the eye of the viewer. People enjoy interacting with attractive design, not simply with web development. Everything generated for marketing should be of excellent quality in order to boost user engagement and conversions. As a result, having a great web design is essential for any website or app.

A user can be held captive for a considerable amount of time by an interesting user interface. As a result, a user interface should be both straightforward and distinct. This contributes to a positive user experience. So, with the little resources I had, I had to build a user interface that was both basic and unique. This contributes to a positive user experience.

We can attract new clients and retain existing ones by implementing an appealing design. The user interface (UI) is the most fundamental part of web development and can serve as a link between the user and the online applications. As a result, I needed to create an engaging user interface with straightforward navigation, easy-to-read text, and appealing images. To produce an appealing design, I had to use creativity and a thorough understanding of UI/UX design, coding languages, and developer experiences.

Users will not wait long for a website to load when working in a hospital since there are so many activities to accomplish. If they wait too long, they will just move on to another search result or give up. As a result, the poor loading time is the most significant technical difficulty I've encountered, and it has the potential to seriously harm my firm. By optimizing my code (including the removal of spaces, commas, and other unneeded characters), I can significantly boost the page speed of my website. Remove any code comments, formatting, and unneeded code as well.

Every time a page redirects to another, our visitor must wait for the HTTP request-response cycle to complete. As a result, I tried to condense all of the contents into a single page, however this reduced the appeal and made the website harder to use. So I had to find the balance between the website's attractiveness and its speed while making multiple pages.

In the web development industry, there are numerous frameworks, programming languages, and tools for creating web apps. Choosing the correct tools and frameworks may appear simple, yet it was a significant challenge for me.

Complex tools may be required to solve complex problems. Simple web development jobs, such as developing a blog, can, however, be completed utilizing systems such as WordPress. However, since the client requested that I had to create a complex management system for the hospital on a limited budget, I needed to find a better framework to complete the project. It was also tough to find an open-source framework to use for this project.

Security is one of the most pressing concerns for modern enterprises with a significant internet presence. It is critical to secure the application in order to protect sensitive information. There are numerous methods for increasing the security of a website or app. During the development phase, I considered all available security precautions. Having SSL certificates is one part of global security. It enables the browser and server to communicate in an encrypted manner. This can help to ensure the security of sensitive data such as credit and debit card numbers, transaction details, usernames, and passwords. However, I was unable to appeal for an SSL certificate.

Using multi-factor authentication to secure accounts would also be a good idea. Users would have to go through a series of processes to prove their identity and gain access to their accounts. Applying multi-factor authentication is challenging for me and takes a long time to develop. As a result, I had to start with a basic authentication.

Task 4

4.1 The Importance of Quality Assurance in the Web Development Process

4.1.1 Introduction

Quality assurance is an essential component of professional website creation that enables a business owner to obtain a high-quality website. Website QA (Quality Assurance) is the process of testing a website to find flaws, errors, or oversights that may have gone undetected during web design and development before going live.

A website or web app is the business platform for any online firm. A website can be highly logical and attractively designed, but it will lose users if it does not perform as intended or is loaded with annoying bugs. Website Quality Assurance (QA) ensures that when consumers use a web app or website, they have a functional user interface and the greatest user experience possible. It inspects a website or online application for problems that may have been overlooked during design and development.

Quality assurance generally involves:

1. Requirement testing
2. Design evaluation
3. Functionality testing

1) Requirement Testing :

Requirement testing considers the client's specified needs as well as additional client-approved design documents. The QA tester creates a list of expectations and needs, which he or she uses to evaluate the layout, prototype, and/or alpha version of a website or web app. Throughout the development process, requirement testing is critical to ensuring that the project team fulfills all of their mission objectives. Since my client asked me to provide an efficient system for managing hospital records in a cost-effective manner, all the given requirements were considered under this testing criteria.

2) Design Evaluation :

Design testing examines the visual design of the web app or website. The QA tester analyzes whether the look and feel of the web app or website are consistent with the proposed design and overall layout when testing the design. Design testing is a multifaceted procedure that involves tasks such as browser testing and mobile testing.

The designed wireframes and customer requirements were evaluated. Then these wireframes were shown to the client and asked for feedback from them. The initial system was then designed as a prototype, and it was shown to the client how it worked before the full system was completed.

3) Functionality Testing :

Functionality testing assesses a web app's or website's UI (User Interface) and UX (User Experience). It is usually the last step in the Quality Assurance testing process. It is a thorough process that guarantees every link, button, form, and other element works properly. The website or app is tested across several browsers and platforms to ensure that the end user has no trouble navigating or utilizing the site or app.

As a result, the prototype I created for the system as an initial design was tested and shown to the customer. Then I asked for the customer's feedback before moving into the final product design. Then the functionality, design, and every other aspect were thoroughly tested.

So before giving the final output to the client, the client's requirements were tested, then the design was tested, and also the functionality was tested. So this helps to ensure the quality of the web site before it goes live and before it is given to the customer.

4.1.2 Types of QA Testing

There are various forms of QA testing that may be used to develop a consistent QA procedure.

1) Functional testing

During the functional testing phase, we check to verify if the site functions properly by evaluating buttons, forms, menus, and links. Additionally, ensure that any form submissions are routed to the appropriate persons and that any automatic answers are properly configured.

As a result, I tested every button, form, menu, and link in my system as a functional test. Also, I double-checked all form submission routes to verify they were routed to the correct persons, and I double-checked those automated answers were properly configured.

2) Performance testing

When it comes to website quality testing, one of the most critical phases is performance testing. We would test page-load speed, fundamental web vitals, and how the site responds to attempted overloads at this period. Assessing the impact of a rapid rise in site traffic, for example, is known as a spike test.

As a result of this performance testing, I was able to examine the system's behavior under heavy stress. I examined several sections of the product under various loads; performance testing assist establish the product's reliability, stability, and availability, as well as if increasing load may damage system performance.

3) User Testing

User testing guarantees that visitors can browse the site logically and obtain the information they need to support them and take the required action. I looked at factors like content hierarchy, site search functionality, and menu structure during user testing.

4) Compatibility testing

Compatibility testing ensures that our website works properly in a variety of browsers, including Chrome, Safari, and Firefox, on both desktop and mobile devices. This ensures that text and graphics line and stack correctly at different breakpoints for different screen widths.

5) Security Testing

Security testing will enable us to identify and fix weaknesses on the site, as well as protect it from potential attacks. As we progress through this phase, we will look at logins, restricted sections, payment gateways, and third-party interfaces. While ensuring that our chosen content management system (CMS), whether Drupal, WordPress, or another service, is up to date on security patches.

6) SEO Testing

Search Engine Optimization (SEO) testing determines whether the website is optimized to be discovered on search engines such as Google, Bing, and Yahoo. During this procedure, we would search for both technical and on-page SEO problems, such as meta descriptions, alt tags, page titles, and SEO optimized URL structures. As we go through this process, we need to keep an eye out for both technical and on-page SEO mistakes, such as those involving meta descriptions, alt tags, page names, and SEO-friendly URL structures.

7) Code Review

The source code of the website will be inspected at this stage of the QA process to detect and rectify any mistakes that may have been missed during the initial development phase.

8) Unit Testing

Unit testing is a type of ground-level testing that is carried out to test individual units or functionality of the software under development. Unit testing helps to understand the core functioning of the code and tests the likelihood of any change generating a fault, which can then be rapidly recognized and repaired.

9) Smoke Testing

Smoke testing includes examining the application's basic operation to see if important system features are functioning as planned. Smoke tests, which are especially beneficial after a fresh build, assist in assessing whether more expensive tests on the application in the newly deployed environment are required. They also help to determine whether faults are severe enough to cause a proposed system release to be rejected.

10) User Acceptance Testing

User acceptance tests are the final test in the QA lifecycle and are performed immediately before the final release to ensure that the product or application can handle real-world circumstances. It verifies if the system meets business requirements by reproducing user behavior and rejects changes if specific requirements are not met.

As I previously stated, I performed user acceptance soon after creating the wireframes, first design, and final product. I questioned the client if this system met all of their requirements and if anything, else needed to be added.

4.1.3 Tools to use in QA

Manual QA can be a difficult task. As I have stated, we must automate where practical and required. It is simple to test our site on our browser and PC and conclude that it will look wonderful and work as expected across all browsers and devices, and that users would intuitively know what to do. We may need to employ a variety of tools to test various stages of our QA process. Here are some tools that we can utilize to help us with your QA testing.

1) TestRail

TestRail is entirely web-based testing software that provides real-time insights to assist us in tracking our progress. We can use it online or download it. The tool interfaces readily with test automation technologies.

2) CrossBrowserTesting

This cloud-based application is used to run tests on mobile and desktop browsers. There are around 2050 browsers available for testing. It'll be useful for compatibility and regression testing. It includes a free trial.

3) Web Developer Form Filler

we use forms on our site, even if they are as simple as a contact form, we must test whether the form only accepts the specified type of user input, whether it submits the correct information, and even whether the information shared (especially if it is personal user information) is displayed on the browser URL. The Web Developer Form Filler tool allows us to test the operation of a form. All we have to do is enter the values we want in your input fields, and the tool will handle the rest for us. This allows us to test our forms more quickly.

4) Ranorex Web test

This tool is perfect for evaluating user interfaces across browsers and operating systems (Mac, Android, Windows, and Linux). We can run all of the tests on our own PC or desktop with Ranorex. It includes a free trial period and built-in test reports.

5) Window Resizer

Window Resizer is a Chrome Extension that allows us to see how our site looks in various window widths. This application allows us to evaluate the most regularly used desktop and mobile sizes. Custom screen sizes can also be added. The program advises us on what content should be displayed above the fold, as well as how to make our screens look better on various screen sizes.

6) SEOptimer

This tool is useful for SEO testing. It does a site audit to assist us determine what we are doing correctly in terms of SEO and what needs to be improved in order for our site to rank higher in search engine results. Even though we lack SEO skills, it is straightforward to utilize. It provides us with an update on the success of our website and helps us to customize our report to our brand. We also get to choose the language in which our report will be written.

4.1.4 How QA helped to build the Hospital Management System

This helped me in presenting my brand as credible. When everything on the planned site functions properly, there is a significant likelihood that customers would associate with our Apex organization with quality. As a result, they will want to purchase our products, pay for our services, and be connected with our company's brand. And QA testing assisted me in identifying issues that could have serious ramifications, such as a critical security vulnerability.

And, in the end, QA testing helped me save the company money and time that would have been wasted resolving errors long after the site had been published. It is also less expensive to resolve a bug before the site is launched rather than later. As a result, no users will be impacted. Because we won't need your customer service to explain to users why the site isn't working as planned, and I won't have to 'shut down the site' for fixes, losing traffic and money.

4.2 Documentation for developed System

4.2.1 Why system documentation

All written documentation and resources dealing with the development and use of a system/software product. Whether developed by a small team or a large business, all web design development products require some form of documentation. Documentation exists to describe product functionality, organize project-related data, and allow stakeholders and developers to discuss any significant questions that arise. The major goal of good documentation is to make sure that developers and stakeholders are on the same page in order to achieve the project's goals.

There are two basic types of documentation for software products.

1. User Documentation - User documentation refers to documents written primarily for product end-users and system administrators. Tutorials, user guides, troubleshooting manuals, installation, and reference manuals are all examples of user documentation.
2. Technical Documentation - This document contains information about the system as well as its technological components. It includes documents such as requirements, design decisions, architecture descriptions, program source code, and help guides.

4.3 User Documentation

4.3.1 What is user documentation

End-user documentation, often known as user documentation, refers to the documentation provided to end-users for a product or service. Its purpose is to provide product information. Also known as "user help," this is a term that is frequently used. It could be based on a delivered product or service, or a complete end-to-end project documentation that directly benefits the end-user.

Technically, a useful user document should have all relevant documentation and necessities for the whole product life cycle. The user documentation is part of the end-to-end product that is supplied to the consumer.

4.3.2 Types of User Documentation

- 1) Description Document :- The Product Description Document provides a full summary of the product, including all of the services it provides. End-users read this page and decide whether or not this is the product they are looking for.
- 2) Installation and Setup :- Provides thorough instructions on how to install and set up the product, as well as how to use it.
- 3) Product or User Manual :- It explains the product's basic functionalities with illustrations. It contains all of the 'How-to' instructions for using the product on a regular basis. We can write a fantastic Product handbook in no time.

4.3.3 User Documentation for Arogya Hospital Management System

Contents

- 1) System Description
- 2) System References
- 3) System Features
- 4) System Requirements
- 5) User Interfaces

1) System Description

This system is made for Administrators in the Arogya Hospitals to use. I've provided login credentials to access the system hence it enhances the security of the system. For admin this system is providing to store details of staff, patients, medicines and other stuff. And also, with this system admin can classify the staff types separately. Not only that, but also admin can store details about wards and room availability of each ward. Admin can do that easily by looking at the stored data in the database.

2) System References

I used latest platforms, plugins and technologies when creating this system

- Bootstrap V5.1
- Sublime Text
- MySQL (PHP My Admin 5.1.1)
- WAMP Server 3.2.6 – 64bit

3) System Features

This system is coming with small file containing the database base, source code along with the setup. User can easily install the setup by double clicking the “SETUP.exe” file to any location user want.

4) System Requirements

This system can be installed to any basic computers which having Windows Operating System.

5) User Interfaces

Login Page →

After loading the system 1st thing all users can see is the login form. User must select the job role they are having in the combo box. Then user should give login credentials in the Username and Password fields which system developers have given separately. After entering login credentials user can simply enter to the system by clicking Login button. If user want to erase given data, user can do it by clicking on the clear button. If user want to exit, he/she can do it by clicking Exit button.

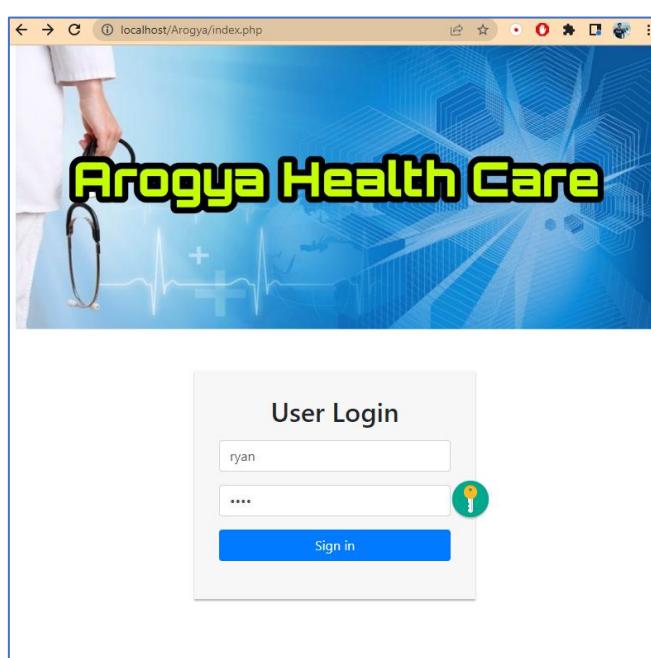


Figure 4. 1 Login Page interface

Home Page →

- After giving correct login credential user directed to home page.
- User can surf through this interface and can give email to subscribe our newsletters (Can unsubscribe anytime they want).
- User can navigate details by scrolling down.
- User can navigate through each page by clicking this links on the navigation bar at the top
- User can go back to the top of the page by clicking the arrow mark at the footer.

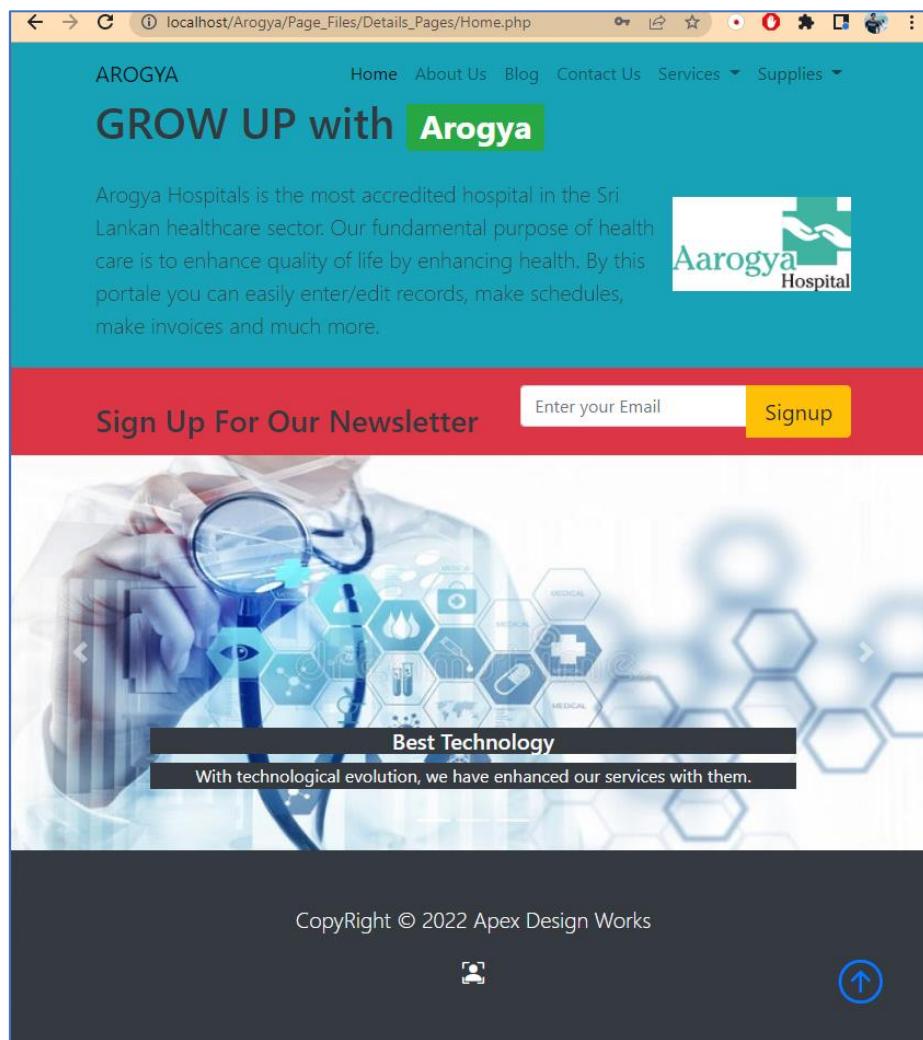


Figure 4. 2 Home Page interface

About Us Page →

- User can navigate details by scrolling down.
- User can navigate through each page by clicking this links on the navigation bar at the top
- User can go back to the top of the page by clicking the arrow mark at the footer.

The screenshot shows the 'About Us' page of a website. At the top, there is a navigation bar with the logo 'AROGYA' and links for Home, About Us, Blog, Contact Us, Services, and Supplies. Below the navigation bar, there are three dark blue boxes with white text and icons. The first box is titled 'Awarded' with a trophy icon, stating: 'We have been recognized for our exceptional service and care by both international and local awarding bodies.' It has a 'Read More>' button. The second box is titled 'Secured' with a shield icon, stating: 'Your health is our priority. We give our priority to helping you, healing you, and protecting you.' It also has a 'Read More>' button. The third box is titled 'Excellent Staff' with a people icon, stating: 'We have an expert medical staff to provide the best health recovery possible. So don't worry about your health.' It has a 'Read More>' button. Below these boxes, the page title 'Our Specialists' is displayed in a large teal section. Underneath the title, it says: 'Our Specialists all have remarkable working experience with international recognition.' Two profiles are shown: 'Prof. Shaman' with a photo of a man and 'Prof. Jithangi' with a photo of a woman.

Figure 4. 3 About Us Page interface

Blog Page →

- User can navigate details by scrolling down.
- User can navigate through each page by clicking this links on the navigation bar at the top.
- User can get more details about relevant blog by clicking on “Read More” buttons.
- User can go back to the top of the page by clicking the arrow mark at the footer.

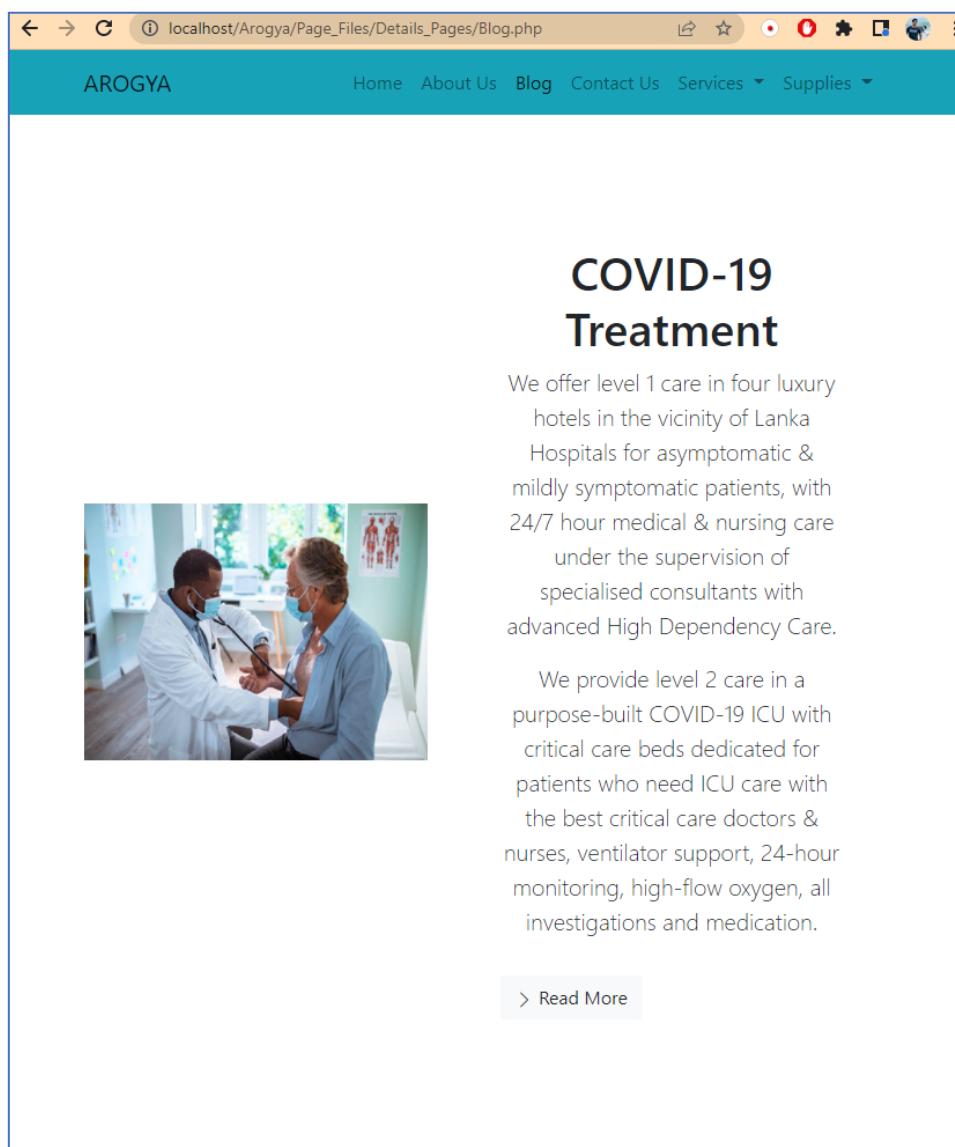


Figure 4. 4 Blog Page interface

Contact Us Page →

- User can navigate details by scrolling down.
- User can navigate through each page by clicking this links on the navigation bar at the top.
- User can view the answers for each frequently asked question by clicking on the question.
- User can zoom in and zoom out by using scroll button in the mapbox.
- User can go back to the top of the page by clicking the arrow mark at the footer.

The screenshot shows a web browser window with the URL localhost/Arogya/Page_Files/Details_Pages/Contact.php. The page has a teal header bar with the logo 'AROGYA' and navigation links: Home, About Us, Blog, Contact Us, Services, and Supplies. Below the header is a section titled 'Frequently Asked Questions' with a collapsed panel for 'What procedures will I have in the Emergency Department?'. The expanded content for this question states: 'Treatments and procedures may include lab work, radiological studies, CAT scans, ultrasound, EKG for cardiac review, medication administration and physician evaluations.' Below this are three more collapsed questions: 'How is the Emergency Department staffed?', 'Will I be allowed to have visitors during my treatment?', and 'What happens if I am admitted?'. To the right of the FAQ section is a map showing the location of the South Asian Institute of Technology and Faculty of Computing in Pittugala, with Lake Road and E002 road visible. At the bottom of the page is a dark footer bar with the text 'CopyRight © 2022 Apex Design Works' and icons for a person and an upward arrow.

Figure 4.5 Contact Us Page interface

Staff Management Page →

- User can go to this form by clicking on the Services tab on the navigation bar and then selecting the “Staff” link.
- To save new records user must click on “Create New User” button. User can simply save new records by filling the text box fields and hitting the Submit button.
- When updating a specific record, user should click Edit button on the related row. User can make changes on the fields then by hitting the Edit button the record will be updated.
- When deleting a specific record, user should click on the Edit button on the related row.
- User can go back to home page by clicking the Back button at the bottom or through navigation bar at the top.
- User can exit from the system by click on the logout button.
- User can navigate through each page by clicking this links on the navigation bar at the top.

The screenshot shows a web browser window with the URL `localhost/Arogya/Page_Files/Staff.php`. The page has a header with the AROGYA logo and a navigation menu with links for Home, About Us, Blog, Contact Us, Services, and Supplies. The main content area is titled "Staff Management". It features a table with the following data:

| ID Number | Name | Type | Options | | |
|-----------|--------------------|------------|----------------------|----------------------|------------------------|
| 1 | Pavel Fernandoooo | Doctor | View | Edit | Delete |
| 2 | Ryan Wickramaratne | Nurse | View | Edit | Delete |
| 3 | Judith Michelle | Technician | View | Edit | Delete |
| 4 | Kyle Wickramaratne | Attendant | View | Edit | Delete |
| 5 | Sadew Fernando | Doctor | View | Edit | Delete |
| 6 | Mashi Perera | Nurse | View | Edit | Delete |
| 7 | Shyni Perera | Attendant | View | Edit | Delete |

A yellow "Back" button is located at the bottom right of the page.

Figure 4. 6 Staff Management Page interface

Patient Management Page →

- User can go to this form by clicking on the Services tab on the navigation bar and then selecting the “Patients” link.
- To save new records user must click on “Create New User” button. User can simply save new records by filling the text box fields and hitting the Submit button.
- When updating a specific record, user should click Edit button on the related row. User can make changes on the fields then by hitting the Edit button the record will be updated.
- When deleting a specific record, user should click on the Edit button on the related row.
- User can create new invoice by clicking on the “Patient Invoice” button at the bottom.
- User can exit from the system by click on the logout button.
- User can go back to home page by clicking the Back button at the bottom or through navigation bar at the top.
- User can navigate through each page by clicking this links on the navigation bar at the top.

| ID | Name | Age | Ward | Options | | |
|----|--------------------|-----|--------|---------|------|--------|
| 1 | Judith Michelle | 20 | Ward 2 | View | Edit | Delete |
| 2 | Ryan Wickramaratne | 25 | Ward 1 | View | Edit | Delete |
| 3 | Pavel Fernando | 15 | Ward 3 | View | Edit | Delete |
| 4 | Malshan Fernando | 25 | Ward 2 | View | Edit | Delete |

Patient Invoice

Back

Figure 4. 7 Patient Management Page interface

Wards Management Page →

- User can go to this form by clicking on the Services tab on the navigation bar and then selecting the “Wards” link.
- To edit specific ward user must click on the relevant row’s “Manage the Ward” button. User can simply save new updated records by filling the text box fields and hitting the Submit button.
- User can't delete wards.
- User can go back to home page by clicking the Back button at the bottom or through navigation bar at the top.
- User can exit from the system by click on the logout button.
- User can navigate through each page by clicking this links on the navigation bar at the top.

| Ward Number | Doctor 1 | Doctor 2 | Nurse 1 | Nurse 2 | Attendant 1 | Attendant 2 |
|-------------|--------------------|----------------|--------------------|--------------|--------------------|--------------|
| Ward 1 | Pavel Fernandooooo | Sadew Fernando | Ryan Wickramaratne | Mashi Perera | Kyle Wickramaratne | Shyni Perera |
| Ward 2 | Pavel Fernandooooo | | Ryan Wickramaratne | | Kyle Wickramaratne | |
| Ward 3 | Sadew Fernando | | Mashi Perera | | Shyni Perera | |
| Ward 4 | Pavel Fernandooooo | | Mashi Perera | | | |

Figure 4. 8 Wards Management Page interface

Rooms Management Page →

- User can go to this form by clicking on the Services tab on the navigation bar and then selecting the “Rooms” link.
- To edit specific ward’s room details user must click on the relevant row’s “Manage the Ward Rooms” button. User can simply save new updated records by filling the text box fields and hitting the Submit button.
- User can’t delete wards.
- User can go back to home page by clicking the Back button at the bottom or through navigation bar at the top.
- User can exit from the system by click on the logout button.
- User can navigate through each page by clicking this links on the navigation bar at the top.

The screenshot shows a web application interface titled "Room Management". At the top, there is a navigation bar with links for Home, About Us, Blog, Contact Us, Services, and Supplies. A red "Logout" button is visible on the left. Below the navigation bar, the main title "Room Management" is centered. A table titled "Edit Ward Room Status" lists four wards: Ward 1, Ward 2, Ward 3, and Ward 4. Each row contains a "Manage the Ward Rooms" button. Below this table, a section titled "Ward Rooms Details" displays a grid of room availability for each ward. The grid has columns for Ward Number, Room 1, Room 2, Room 3, and Room 4. The data is as follows:

| Ward Number | Room 1 | Room 2 | Room 3 | Room 4 |
|-------------|-----------|-----------|-----------|-----------|
| Ward 1 | No | No | Available | Available |
| Ward 2 | Available | Available | Available | Available |
| Ward 3 | No | No | No | Available |
| Ward 4 | No | No | No | No |

A yellow "Back" button is located at the bottom center of the page.

Figure 4. 9 Rooms Management Page interface

Medicine Stock Management Page →

- User can go to this form by clicking on the Supplies tab on the navigation bar and then selecting the “Medicines” link.
- To save new records user must click on “Add New Medicine” button. User can simply save new records by filling the text box fields and hitting the Submit button.
- When updating a specific record, user should click Edit button on the related row. User can make changes on the fields then by hitting the Edit button the record will be updated.
- When deleting a specific record, user should click on the Edit button on the related row.
- User can go back to home page by clicking the Back button at the bottom or through navigation bar at the top.
- User can exit from the system by click on the logout button.
- User can navigate through each page by clicking this links on the navigation bar at the top.

The screenshot shows a web browser window with the URL `localhost/Arogya/Page_Files/Medicines.php`. The page title is "AROGYA". The navigation bar includes links for Home, About Us, Blog, Contact Us, Services, and Supplies. The main content area is titled "Medicine Stock Management". It features a table with the following data:

| ID Number | Name | Type | Quantity | Options | | |
|-----------|--------------|--------------|----------|---------|------|--------|
| 1 | Panadol | Antibiotics | 100 | View | Edit | Delete |
| 2 | Paracetamo | Pain Killers | 500 | View | Edit | Delete |
| 3 | Cough Syrup | Other | 50 | View | Edit | Delete |
| 4 | Gastic Syrup | Other | 50 | View | Edit | Delete |

A yellow "Back" button is located at the bottom left of the content area.

Figure 4. 10 Medicine Stock Management Page interface

Other Supplies Management Page →

- User can go to this form by clicking on the Supplies tab on the navigation bar and then selecting the “Other” link.
- To save new records user must click on “Add New Item” button. User can simply save new records by filling the text box fields and hitting the Submit button.
- When updating a specific record, user should click Edit button on the related row. User can make changes on the fields then by hitting the Edit button the record will be updated.
- When deleting a specific record, user should click on the Edit button on the related row.
- User can go back to home page by clicking the Back button at the bottom or through navigation bar at the top.
- User can exit from the system by click on the logout button.
- User can navigate through each page by clicking this links on the navigation bar at the top.

The screenshot shows a web browser window with the URL `localhost/Arogya/Page_Files/Other.php`. The title bar says "AROGYA". The navigation bar includes links for Home, About Us, Blog, Contact Us, Services, and Supplies. The main content area has a title "Other Supplies Management". Below it are two buttons: "Add New Item" (white background) and "Logout" (red background). A table lists four items:

| ID Number | Name | Type | Quantity | Options | | |
|-----------|------------|---------------------|----------|---------|------|--------|
| 1 | Pen | Office Stationary | 20 | View | Edit | Delete |
| 2 | Bed | Beds and Metrresses | 5 | View | Edit | Delete |
| 3 | Metrresses | Beds and Metrresses | 10 | View | Edit | Delete |
| 4 | Van | Vehicles | 2 | View | Edit | Delete |

A "Back" button is located at the bottom left of the content area.

Figure 4. 11 Other Supplies Management Page interface

4.4 Test Plan

4.4.1 Introduction

Many terms are used to describe software quality assurance. People in this field are frequently flooded with diverse software terminology and are sometimes confused as to what these various expressions imply. For example, the terms "test plan" and "test case" are frequently used interchangeably.

A test plan is a detailed document that outlines the major actions related with a specific testing project. A test case, on the other hand, is simply intended to test a certain situation. A testing strategy should include the following elements:

- The project's scope
- Objectives and goals
- Market segmentation
- Assumptions
- Dates for the beginning and finish of the testing cycle
- Any significant roles, duties, or total resources
- Environment for testing
- Deliverables
- Major dangers and how to manage them
- Reporting and mitigating problems
- The deadline for testing

4.4.2 Test Plan for the Arogya Hospital Management System

Table 4. 1 Test Plan for the Arogya Hospital Management System

| Test Plan | |
|------------------------|---|
| Introduction: | <p>The primary goal of the provided Test Plan is to test the developed website system for Arogya Hospital before to delivering the final project result to them. This document provides instructions for determining whether or not each system function is functioning properly.</p> <p>The purpose of the testing web system is to validate the most important features using the normal procedures for its use. All of the functions that must be tested are listed under the Test to be performed heading.</p> <p>During testing, the difference, various types, and methodologies will be used. The testing will be based on technical requirements.</p> |
| Tests to be performed: | <p>Login Page:- Hide the password characters.</p> <p>Login Page:- Prevent unauthorized users to login to the system.</p> <p>Login Page :- Allowing admin to access to home page by giving correct login credentials.</p> <p>Home Page :- Submitting email to database by filling Sign Up for newsletter button.</p> <p>Home Page :- Navigation Bar functionality.</p> <p>Website responsiveness.</p> |

| | |
|--|--|
| | <p>About Us Page :- Getting lecturers' social media usernames or contact details.</p> <p>Footer's positioning arrow icon functionality.</p> <p>Blog Page :- Read more button functionality.</p> <p>Blog Page :- Read more button functionality.</p> <p>Contact Us Page :- Accordion functionality.</p> <p>Contact Us Page :- Map box functionality.</p> <p>Staff Page :- Create New User button functionality.</p> <p>Staff Page :- Insert New user record to database.</p> <p>Add Staff Details Page :- Popup notification message when submitting records into database.</p> <p>Add Staff Details Page :- Checking for validation.</p> <p>Add Staff Details Page :- Patient Invoice data insert limitation.</p> <p>Staff Page :- View button functionality.</p> <p>Staff Page :- Edit button functionality.</p> <p>Staff Page :- Edit and insert user record to database.</p> <p>Staff Page :- Delete user record from database.</p> |
|--|--|

| | |
|--------------------------------|---|
| | <p>Staff Page :- Back button functionality.</p> <p>Ward Page :- Manage the ward button functionality.</p> <p>Ward Page :- Retrieve data from database to combo box.</p> <p>Ward Page :- Manage the Ward Room button functionality.</p> <p>Room Page :- Logout button functionality.</p> <p>Patient Page :- Patient Invoice data insert.</p> |
| Entry Criteria : | <p>Documentation for testing is being produced.</p> <p>The testing environment is complete.</p> <p>A testing server is available.</p> |
| Test Deliverables: | <p>A list of improvements is required.</p> <p>Checklist should be updated.</p> <p>Report any bugs.</p> |
| Success (Acceptance) criteria: | <p>Putting in place founded solutions from this project to vulnerabilities in the system built for Arogya Hospital.</p> <p>All the checks need to be filled.</p> <p>All the bugs need to be identified.</p> <p>Planed deliverables are ready.</p> |
| Environments for the System: | <p>OS version →</p> <p>Windows 10</p> <p>Mac OS</p> <p>Ubuntu</p> |

| | |
|--------------------------|--|
| | <p>Browsers →</p> <p>Google Chrome</p> <p>Microsoft Edge</p> <p>Mozilla Firefox</p> <p>Internet Explorer</p> <p>Safari for windows</p> <p>For mobile devices →</p> <p>Any mobile devices</p> |
| Risks and Contingencies: | <p>Lack of resources when testing is to begin.</p> <p>Lack of availability required hardware, software, data, or tools.</p> <p>Late delivery of the software, hardware, or tools.</p> <p>Changes to the original requirements or designs.</p> <p>Customer changes requirement before delivery.</p> |

4.5 Test Cases

4.5.1 Introduction

Test cases execute a single test scenario. In general, each scenario has both a positive and negative test case. A test case has two purposes: it is intended to detect problems or bugs in the software application, and it is also intended to demonstrate how the application should be executed if it operates successfully. Furthermore, test cases indicate how the product is used in the actual world and whether it meets the needs of the consumer.

A test case is a written document that includes the following information:

- Data from tests
- Procedures/inputs
- Scenarios
- Definitions
- Environment for testing
- Expected outcomes
- Actual outcomes

4.5.2 Testing Cases according to Test Plan for the Arogya Hospital Management System

In this subtopic I've documented 24 Test Cases according to test plan. Instead of documenting all the test cases of above, I choose unique 24 test cases which is different to each other. I avoided to documenting similar type test cases to save time and space of this documentation.

Ex :- Rather than documenting Edit button testing for all pages I tested only Staff Page's Edit button.

Table 4. 2 Test Case 1

| Test Case 1 | | | |
|-------------------------|--|----------------------|------|
| Function | Login Page:- Hide the password characters. | | |
| Requirement Type | User Requirement | | |
| Inputs | Admin Password : ryan Password : ryan | | |
| Expected Output | **** | Actual Output | **** |
| Result | Success | | |
| Screenshot |  | | |

Table 4. 3 Test Case 2

| Test Case 2 | | | |
|-------------------------|---|----------------------|--|
| Function | Login Page:- Prevent unauthorized users to login to the system. | | |
| Requirement Type | User Requirement | | |
| Inputs | Admin Password : ryan123 Password : ryan123 | | |
| Expected Output | Error Message : | Actual Output | Error Message : Please Enter Correct Username and Password |
| Result | Success | | |
| Screenshot |  | | |

Table 4. 4 Test Case 3

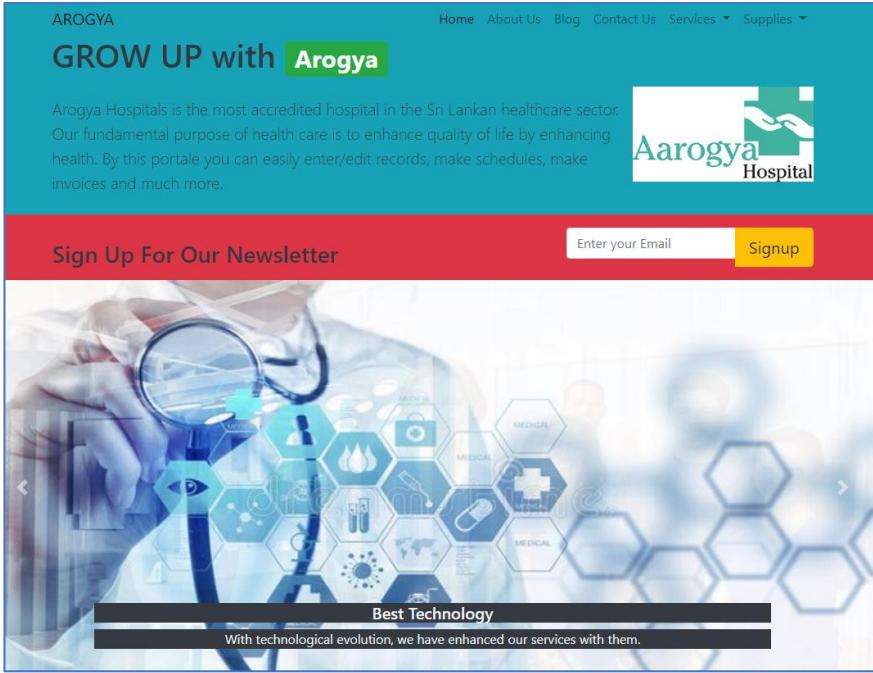
| Test Case 3 | | | |
|-------------------------|--|----------------------|-----------|
| Function | Login Page :- Allowing admin to access to home page by giving correct login credentials. | | |
| Requirement Type | User Requirement | | |
| Inputs | In login page fill the username and password correctly. Then hit “Sign Up” button. | | |
| Expected Output | Home Page | Actual Output | Home Page |
| Result | Success | | |
| Screenshot |  | | |

Table 4. 5 Test Case 4

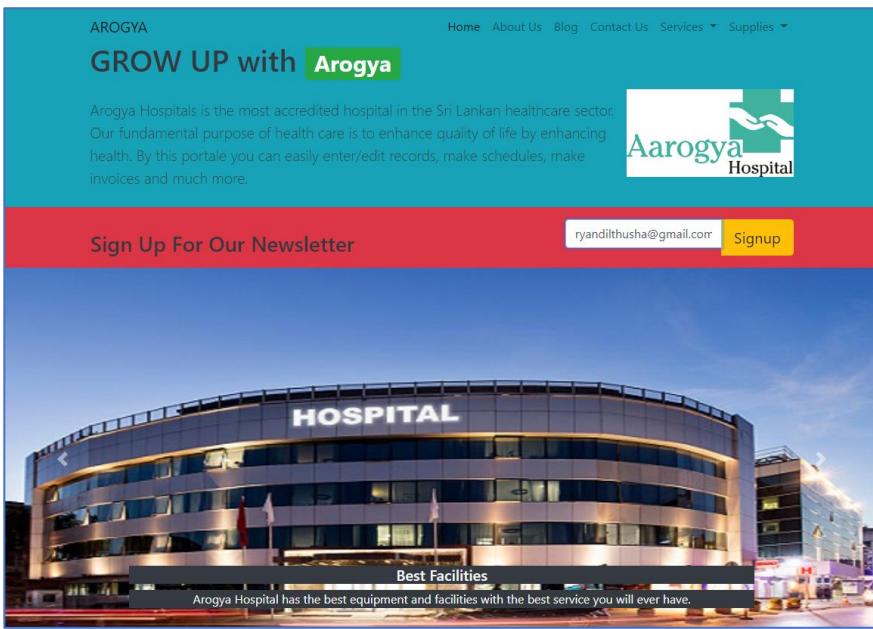
| Test Case 4 | | | |
|-------------------------|--|----------------------|-------------------|
| Function | Home Page :- Submitting email to database by filling Sign Up for newsletter button. | | |
| Requirement Type | System Requirement. | | |
| Inputs | In home page fill the Sign Up for newsletter with an email. Then hit “Sign Up” button. | | |
| Expected Output | Signup Successful message. | Actual Output | Nothing happened. |
| Result | Failed | | |
| Screenshot |  | | |

Table 4. 6 Test Case 5

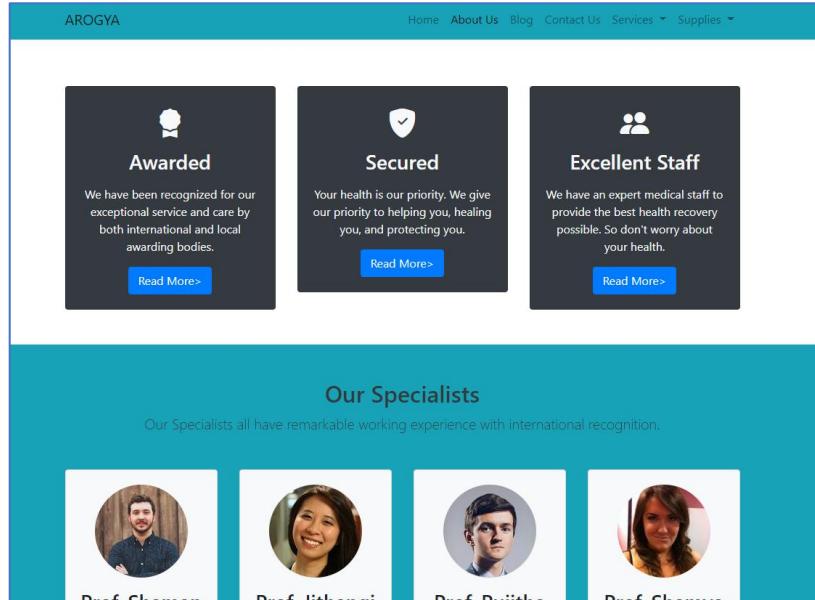
| Test Case 5 | | | |
|-------------------------|---|----------------------|--------------------|
| Function | Home Page :- Navigation Bar functionality. | | |
| Requirement Type | User Requirement. | | |
| Inputs | In home page click on About Us page. | | |
| Expected Output | About Us page open | Actual Output | About Us page open |
| Result | Success | | |
| Screenshot |  <p>The screenshot shows the AROGYA website's homepage. At the top, there is a navigation bar with links: Home, About Us, Blog, Contact Us, Services ▾, and Supplies ▾. Below the navigation bar, there are three dark grey boxes with white icons and text: 'Awarded' (with a trophy icon), 'Secured' (with a shield icon), and 'Excellent Staff' (with a people icon). Each box contains a brief description and a 'Read More' button. Below these boxes, there is a section titled 'Our Specialists' with four circular profile pictures of staff members: Dr. Chamara, Dr. Ishanai, Dr. Dushitha, and Dr. Chamira. Below each profile picture is the name of the staff member.</p> | | |

Table 4. 7 Test Case 6

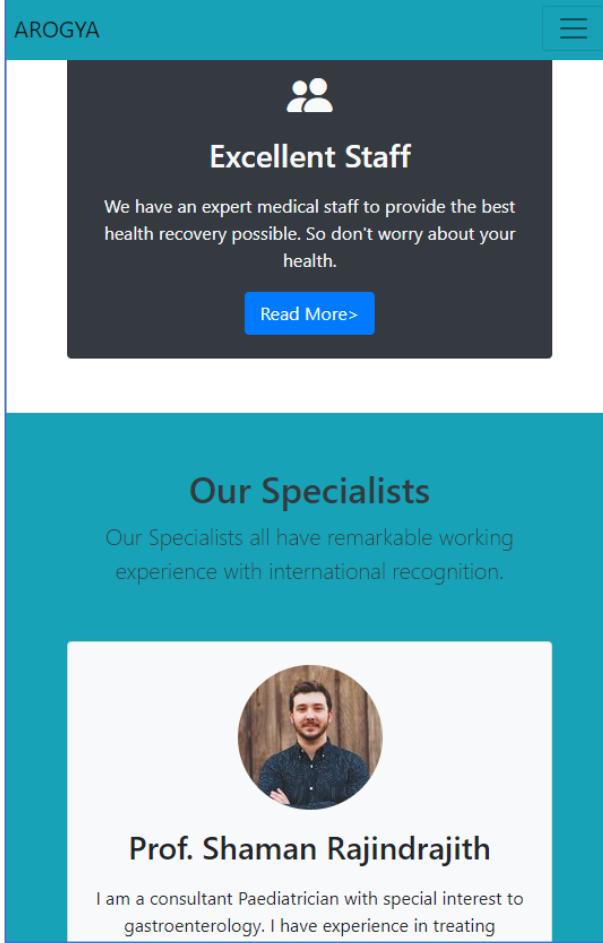
| Test Case 6 | | | |
|-------------------------|---|----------------------|--|
| Function | Website responsiveness. | | |
| Requirement Type | System Requirement. | | |
| Inputs | Resize the browser. | | |
| Expected Output | Warping up page texts and rearranging page content including the navigation bar. | Actual Output | Warping up page texts and rearranging page content including the navigation bar. |
| Result | Success | | |
| Screenshot |  | | |

Table 4. 8 Test Case 7

| Test Case 7 | | | |
|------------------|---|---------------|----------------------|
| Function | About Us Page :- Getting lecturers' social media usernames or contact details. | | |
| Requirement Type | User Requirement. | | |
| Inputs | In about us page click on any social media icon under each lecturer description. | | |
| Expected Output | Getting social media username. | Actual Output | Refreshing the page. |
| Result | Failed | | |
| Screenshot |  | | |

Table 4. 9 Test Case 8

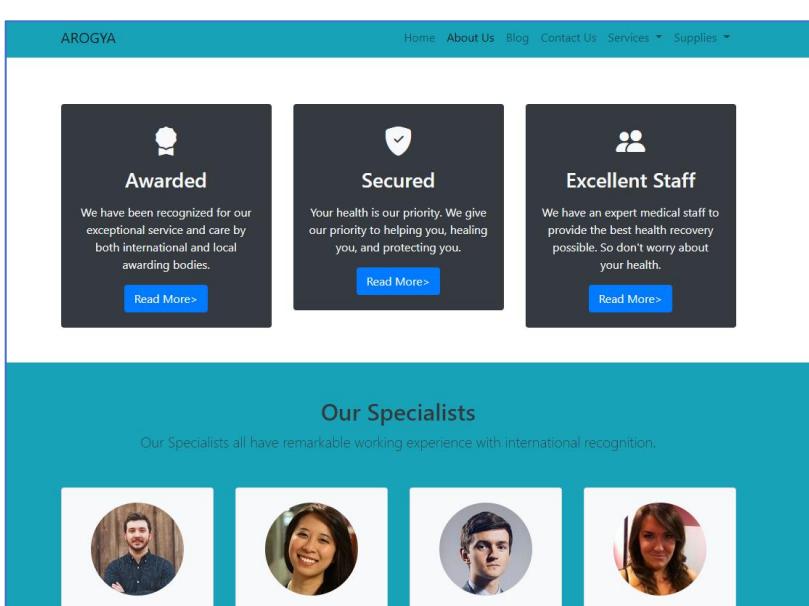
| Test Case 8 | | | |
|-------------------------|---|----------------------|----------------------------|
| Function | Footer's positioning arrow icon functionality. | | |
| Requirement Type | System Requirement. | | |
| Inputs | In home/about us/contact us/blog page footer's positioning arrow | | |
| Expected Output | Direct to top of the page. | Actual Output | Direct to top of the page. |
| Result | Success | | |
| Screenshot |  | | |

Table 4. 10 Test Case 9

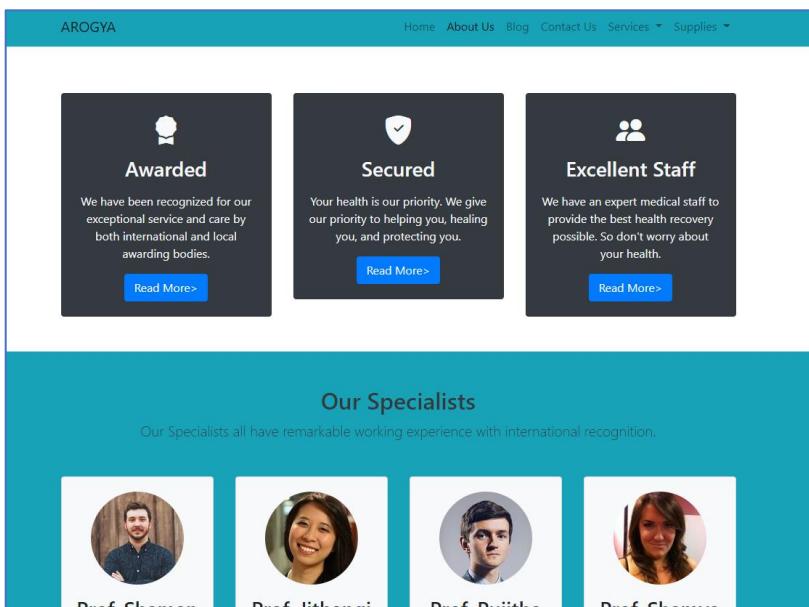
| Test Case 9 | | | |
|-------------------------|---|----------------------|-------------------|
| Function | Blog Page :- Read more button functionality. | | |
| Requirement Type | System Requirement. | | |
| Inputs | In Blog Page click on any blog's "Read More" button. | | |
| Expected Output | Getting more details of the blog. | Actual Output | Nothing happened. |
| Result | Failed | | |
| Screenshot |  | | |

Table 4. 11 Test Case 10

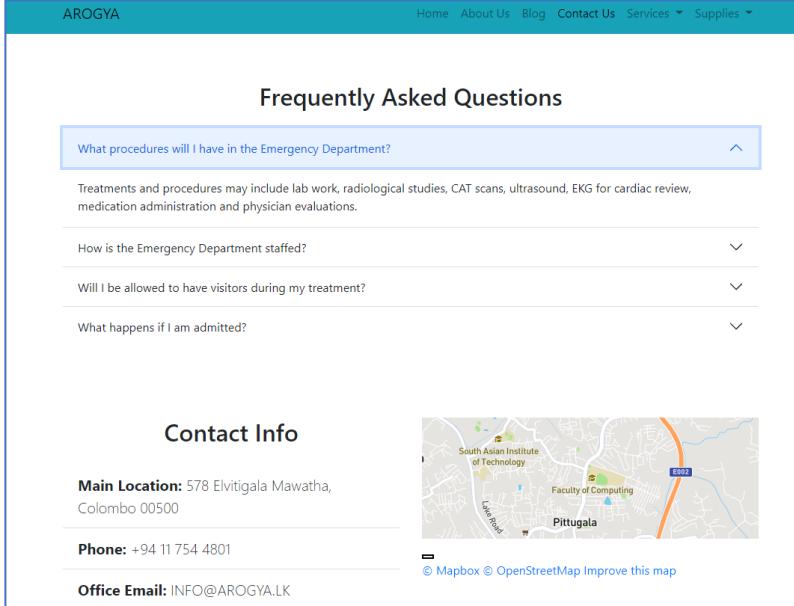
| Test Case 10 | | | |
|-------------------------|---|----------------------|-------------------------------|
| Function | Contact Us Page :- Accordion functionality. | | |
| Requirement Type | System Requirement. | | |
| Inputs | In Contact Us Page clicks on any frequently asked question accordion. | | |
| Expected Output | Getting answers for question. | Actual Output | Getting answers for question. |
| Result | Success | | |
| Screenshot |  <p>The screenshot shows the AROGYA website. At the top, there is a navigation bar with links to Home, About Us, Blog, Contact Us, Services, and Supplies. Below the navigation bar, there is a section titled "Frequently Asked Questions". This section contains several accordions. One accordion is expanded, showing the text: "Treatments and procedures may include lab work, radiological studies, CAT scans, ultrasound, EKG for cardiac review, medication administration and physician evaluations." Below this, there are three collapsed accordions with questions: "How is the Emergency Department staffed?", "Will I be allowed to have visitors during my treatment?", and "What happens if I am admitted?". At the bottom of the page, there is a "Contact Info" section. It includes the main location address ("Main Location: 578 Elvitigala Mawatha, Colombo 00500"), phone number ("Phone: +94 11 754 4801"), and office email ("Office Email: INFO@AROGYA.LK"). To the right of the contact info, there is a map showing the location of the facility.</p> | | |

Table 4. 12 Test Case 11

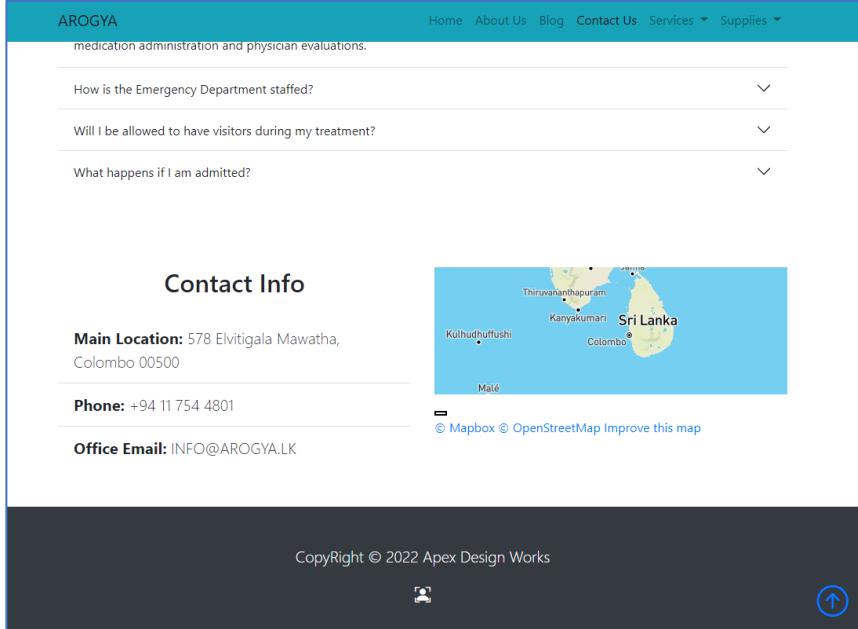
| Test Case 11 | | | |
|-------------------------|---|----------------------|-------------------|
| Function | Contact Us Page :- Map box functionality. | | |
| Requirement Type | System Requirement. | | |
| Inputs | In Contact Us Page scrolls down in map box. | | |
| Expected Output | Zoom out the map. | Actual Output | Zoom out the map. |
| Result | Success | | |
| Screenshot |  | | |

Table 4. 13 Test Case 12

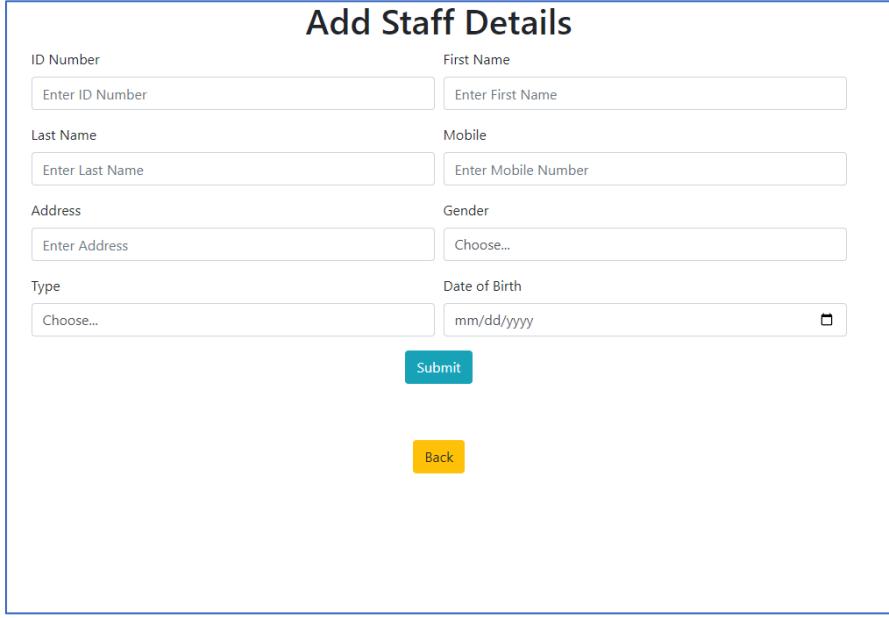
| Test Case 12 | | | |
|-------------------------|---|----------------------|--|
| Function | Staff Page :- Create New User button functionality. | | |
| Requirement Type | System Requirement. | | |
| Inputs | In Staff Page click on Create New User button. | | |
| Expected Output | Directing to “Add Staff Details” page. | Actual Output | Directing to “Add Staff Details” page. |
| Result | Success | | |
| Screenshot |  | | |

Table 4. 14 Test Case 13

| Test Case 13 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------------------------|---|----------------------|--|-----------|------|------|---------|---|----------------|--------|--|---|--------------------|-------|--|---|-----------------|------------|--|---|--------------------|-----------|--|---|----------------|--------|--|---|--------------|-------|--|---|--------------|-----------|--|---|--------------|-------|--|
| Function | Staff Page :- Insert New user record to database. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Requirement Type | User Requirement. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Inputs | In Add Staff Details Page fill the text box fields and then click on Submit button. (In this case I've given Sarah Ashika's details) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Expected Output | Add new details into database and show it in Staff Page's table. | Actual Output | Add new details into database and show it in Staff Page's table. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Result | Success | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Screenshot | <p>The screenshot shows a web application interface titled "Staff Management". At the top, there are buttons for "Create New User" and "Logout". Below the title, there is a table with columns: "ID Number", "Name", "Type", and "Options". The table contains 8 rows of data:</p> <table border="1"> <thead> <tr> <th>ID Number</th> <th>Name</th> <th>Type</th> <th>Options</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Pavel Fernando</td> <td>Doctor</td> <td>View Edit Delete</td> </tr> <tr> <td>2</td> <td>Ryan Wickramaratne</td> <td>Nurse</td> <td>View Edit Delete</td> </tr> <tr> <td>3</td> <td>Judith Michelle</td> <td>Technician</td> <td>View Edit Delete</td> </tr> <tr> <td>4</td> <td>Kyle Wickramaratne</td> <td>Attendant</td> <td>View Edit Delete</td> </tr> <tr> <td>5</td> <td>Sadew Fernando</td> <td>Doctor</td> <td>View Edit Delete</td> </tr> <tr> <td>6</td> <td>Mashi Perera</td> <td>Nurse</td> <td>View Edit Delete</td> </tr> <tr> <td>7</td> <td>Shyni Perera</td> <td>Attendant</td> <td>View Edit Delete</td> </tr> <tr> <td>8</td> <td>Sarah Ashika</td> <td>Nurse</td> <td>View Edit Delete</td> </tr> </tbody> </table> <p>A yellow "Back" button is located at the bottom center of the page.</p> | | | ID Number | Name | Type | Options | 1 | Pavel Fernando | Doctor | View Edit Delete | 2 | Ryan Wickramaratne | Nurse | View Edit Delete | 3 | Judith Michelle | Technician | View Edit Delete | 4 | Kyle Wickramaratne | Attendant | View Edit Delete | 5 | Sadew Fernando | Doctor | View Edit Delete | 6 | Mashi Perera | Nurse | View Edit Delete | 7 | Shyni Perera | Attendant | View Edit Delete | 8 | Sarah Ashika | Nurse | View Edit Delete |
| ID Number | Name | Type | Options | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | Pavel Fernando | Doctor | View Edit Delete | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | Ryan Wickramaratne | Nurse | View Edit Delete | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | Judith Michelle | Technician | View Edit Delete | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | Kyle Wickramaratne | Attendant | View Edit Delete | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | Sadew Fernando | Doctor | View Edit Delete | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | Mashi Perera | Nurse | View Edit Delete | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | Shyni Perera | Attendant | View Edit Delete | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8 | Sarah Ashika | Nurse | View Edit Delete | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 4. 15 Test Case 14

| Test Case 14 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------------------------|--|----------------------|---|-----------|------|------|---------|--|--|---|-------------------|--------|--|--|--|---|--------------------|-------|--|--|--|---|-----------------|------------|--|--|--|---|--------------------|-----------|--|--|--|---|----------------|--------|--|--|--|---|--------------|-------|--|--|--|---|--------------|-----------|--|--|--|---|--------------|-------|--|--|--|---|---------------------|------------|--|--|--|
| Function | Add Staff Details Page :- Popup notification message when submitting records into database. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Requirement Type | User Requirement. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Inputs | In Add Staff Details Page fill the text box fields and then click on Submit button. (In this case I've given Sugath Thilakaratne details) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Expected Output | While adding new details into database show "User Added Successfully" pop up notification message. | Actual Output | Add new details into database. But no popup notification message. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Result | Failed | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Screenshot | <table border="1"> <thead> <tr> <th>ID Number</th> <th>Name</th> <th>Type</th> <th colspan="3">Options</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Pavel Fernandoooo</td> <td>Doctor</td> <td></td> <td></td> <td></td> </tr> <tr> <td>2</td> <td>Ryan Wickramaratne</td> <td>Nurse</td> <td></td> <td></td> <td></td> </tr> <tr> <td>3</td> <td>Judith Michelle</td> <td>Technician</td> <td></td> <td></td> <td></td> </tr> <tr> <td>4</td> <td>Kyle Wickramaratne</td> <td>Attendant</td> <td></td> <td></td> <td></td> </tr> <tr> <td>5</td> <td>Sadew Fernando</td> <td>Doctor</td> <td></td> <td></td> <td></td> </tr> <tr> <td>6</td> <td>Mashi Perera</td> <td>Nurse</td> <td></td> <td></td> <td></td> </tr> <tr> <td>7</td> <td>Shyni Perera</td> <td>Attendant</td> <td></td> <td></td> <td></td> </tr> <tr> <td>8</td> <td>Sarah Ashika</td> <td>Nurse</td> <td></td> <td></td> <td></td> </tr> <tr> <td>9</td> <td>Sugath Thilakaratne</td> <td>Technician</td> <td></td> <td></td> <td></td> </tr> </tbody> </table> | | | ID Number | Name | Type | Options | | | 1 | Pavel Fernandoooo | Doctor | | | | 2 | Ryan Wickramaratne | Nurse | | | | 3 | Judith Michelle | Technician | | | | 4 | Kyle Wickramaratne | Attendant | | | | 5 | Sadew Fernando | Doctor | | | | 6 | Mashi Perera | Nurse | | | | 7 | Shyni Perera | Attendant | | | | 8 | Sarah Ashika | Nurse | | | | 9 | Sugath Thilakaratne | Technician | | | |
| ID Number | Name | Type | Options | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | Pavel Fernandoooo | Doctor | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | Ryan Wickramaratne | Nurse | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | Judith Michelle | Technician | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | Kyle Wickramaratne | Attendant | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | Sadew Fernando | Doctor | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | Mashi Perera | Nurse | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | Shyni Perera | Attendant | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8 | Sarah Ashika | Nurse | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 9 | Sugath Thilakaratne | Technician | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 4. 16 Test Case 15

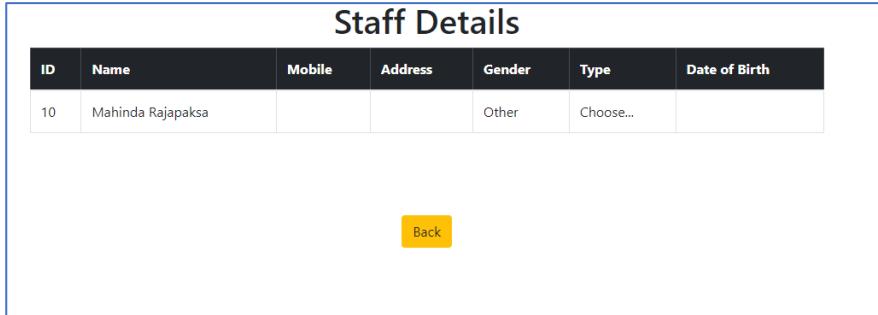
| Test Case 15 | | | |
|-------------------------|--|----------------------|--|
| Function | Add Staff Details Page :- Checking for validation. | | |
| Requirement Type | System Requirement. | | |
| Inputs | In Add Staff Details Page fill some the text box fields (Not all) and then click on Submit button. (In this case I've given Mahinda Rajapaksha details) | | |
| Expected Output | Error message with asking to fill all the details. | Actual Output | Add new details into database. Didn't popup any error message. |
| Result | Failed | | |
| Screenshot |  <p>The screenshot shows a 'Staff Details' page with a table. The table has columns: ID, Name, Mobile, Address, Gender, Type, and Date of Birth. There is one row with data: ID 10, Name 'Mahinda Rajapaksa', Gender 'Other', and Type 'Choose...'. Below the table is a yellow 'Back' button.</p> | | |

Table 4. 17 Test Case 16

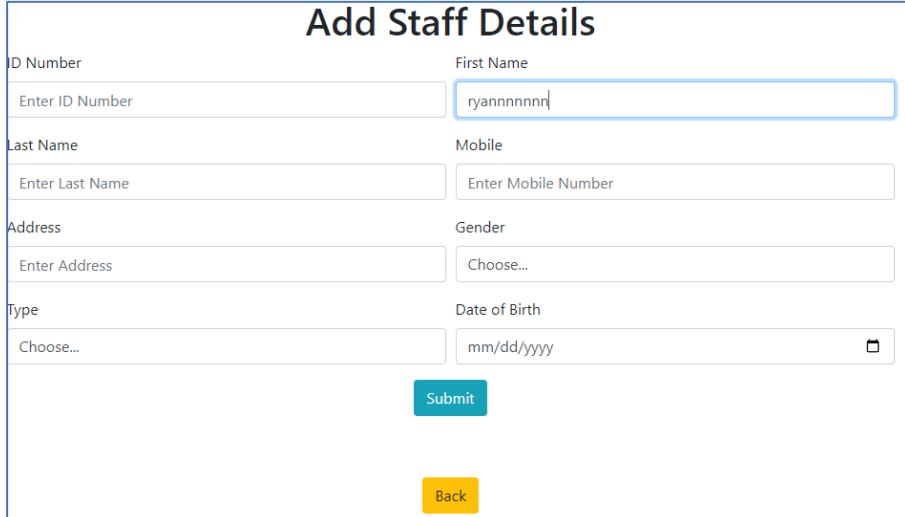
| Test Case 16 | | | |
|-------------------------|---|----------------------|-----------------------------|
| Function | Add Staff Details Page :- Data characters insert limitation. | | |
| Requirement Type | System Requirement. | | |
| Inputs | In Add Staff Details page try to give long characters as input. | | |
| Expected Output | Can't type long characters. | Actual Output | Can't type long characters. |
| Result | Success | | |
| Screenshot |  | | |

Table 4. 18 Test Case 17

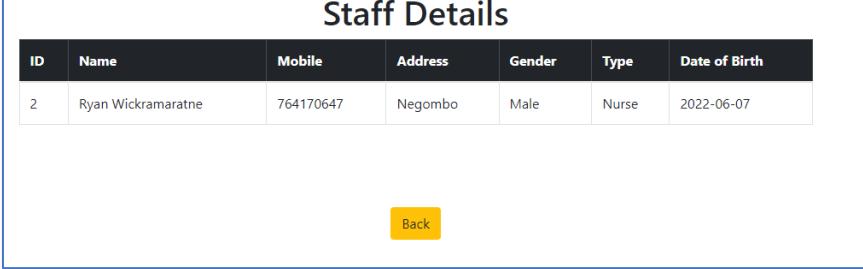
| Test Case 17 | | | | | | | | | | | | | | | | | |
|-------------------------|--|----------------------|---|--------|-------|---------------|---------|--------|------|---------------|---|--------------------|-----------|---------|------|-------|------------|
| Function | Staff Page :- View button functionality. | | | | | | | | | | | | | | | | |
| Requirement Type | User Requirement. | | | | | | | | | | | | | | | | |
| Inputs | In Staff Page click on any row View button. (In this case I've clicked on Ryan Wickramaratne view button) | | | | | | | | | | | | | | | | |
| Expected Output | Directing to “Staff Details” page and show all the details. | Actual Output | Directing to “Staff Details” page and show all the details. | | | | | | | | | | | | | | |
| Result | Success | | | | | | | | | | | | | | | | |
| Screenshot |  <p>The screenshot shows a table titled "Staff Details" with the following data:</p> <table border="1"> <thead> <tr> <th>ID</th> <th>Name</th> <th>Mobile</th> <th>Address</th> <th>Gender</th> <th>Type</th> <th>Date of Birth</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>Ryan Wickramaratne</td> <td>764170647</td> <td>Negombo</td> <td>Male</td> <td>Nurse</td> <td>2022-06-07</td> </tr> </tbody> </table> <p>Below the table is a yellow "Back" button.</p> | | | ID | Name | Mobile | Address | Gender | Type | Date of Birth | 2 | Ryan Wickramaratne | 764170647 | Negombo | Male | Nurse | 2022-06-07 |
| ID | Name | Mobile | Address | Gender | Type | Date of Birth | | | | | | | | | | | |
| 2 | Ryan Wickramaratne | 764170647 | Negombo | Male | Nurse | 2022-06-07 | | | | | | | | | | | |

Table 4. 19 Test Case 18

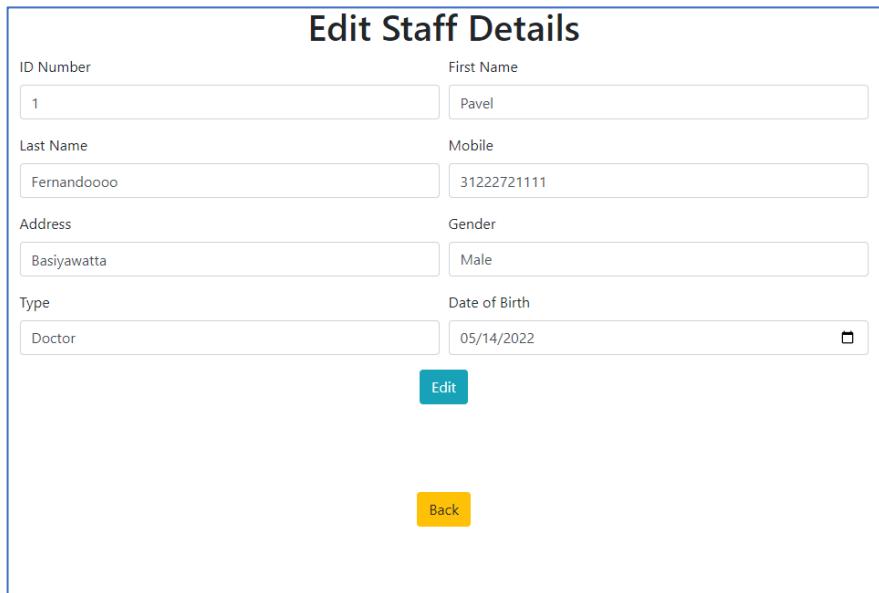
| Test Case 18 | | | |
|-------------------------|---|----------------------|---|
| Function | Staff Page :- Edit button functionality. | | |
| Requirement Type | System Requirement. | | |
| Inputs | In Staff Page click on any rows Edit button. | | |
| Expected Output | Directing to “Edit Staff Details” page. | Actual Output | Directing to “Edit Staff Details” page. |
| Result | Success | | |
| Screenshot |  | | |

Table 4. 20 Test Case 19

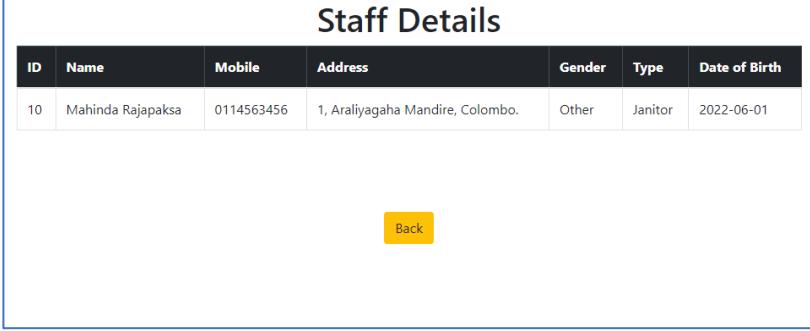
| Test Case 19 | | | | | | | | | | | | | | | | | |
|-------------------------|---|----------------------|---|--------|---------|---------------|---------|--------|------|---------------|----|-------------------|------------|----------------------------------|-------|---------|------------|
| Function | Staff Page :- Edit and insert user record to database. | | | | | | | | | | | | | | | | |
| Requirement Type | User Requirement. | | | | | | | | | | | | | | | | |
| Inputs | In Edit Staff Details Page edit the text box fields and then click on Submit button. (In this case I've edited Mahinda Rajaraksa details) | | | | | | | | | | | | | | | | |
| Expected Output | Edit new details into database and show it in Staff Page's table. | Actual Output | Edit new details into database and show it in Staff Page's table. | | | | | | | | | | | | | | |
| Result | Success | | | | | | | | | | | | | | | | |
| Screenshot |  <p>The screenshot shows a 'Staff Details' page with a table containing one row of data. The table has columns for ID, Name, Mobile, Address, Gender, Type, and Date of Birth. The data in the table is:</p> <table border="1"> <thead> <tr> <th>ID</th> <th>Name</th> <th>Mobile</th> <th>Address</th> <th>Gender</th> <th>Type</th> <th>Date of Birth</th> </tr> </thead> <tbody> <tr> <td>10</td> <td>Mahinda Rajapaksa</td> <td>0114563456</td> <td>1, Araliyagaha Mandire, Colombo.</td> <td>Other</td> <td>Janitor</td> <td>2022-06-01</td> </tr> </tbody> </table> <p>Below the table is a yellow 'Back' button.</p> | | | ID | Name | Mobile | Address | Gender | Type | Date of Birth | 10 | Mahinda Rajapaksa | 0114563456 | 1, Araliyagaha Mandire, Colombo. | Other | Janitor | 2022-06-01 |
| ID | Name | Mobile | Address | Gender | Type | Date of Birth | | | | | | | | | | | |
| 10 | Mahinda Rajapaksa | 0114563456 | 1, Araliyagaha Mandire, Colombo. | Other | Janitor | 2022-06-01 | | | | | | | | | | | |

Table 4. 21 Test Case 20

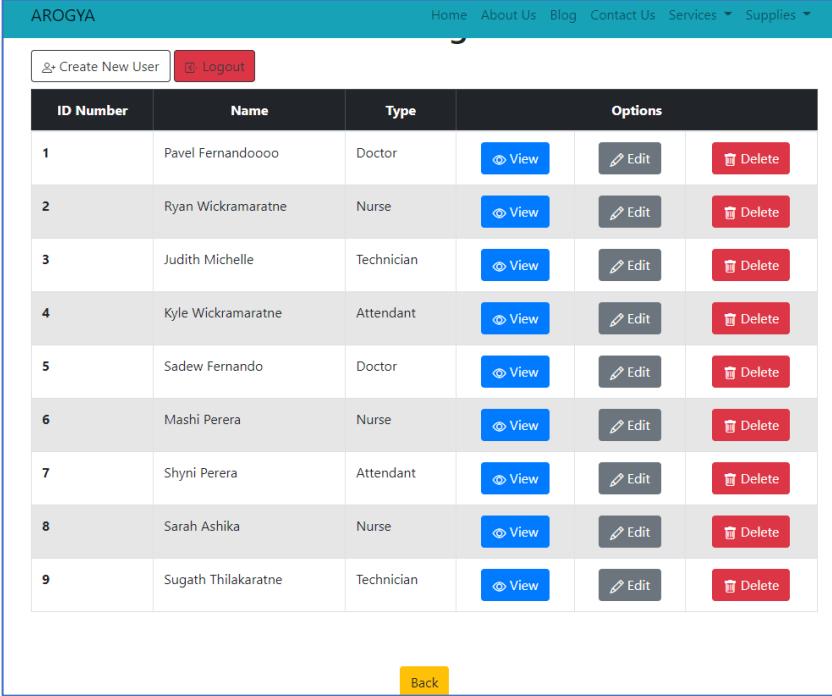
| Test Case 20 | | | |
|-------------------------|--|----------------------|--|
| Function | Staff Page :- Delete user record from database. | | |
| Requirement Type | User Requirement. | | |
| Inputs | In Staff Page click on any row Delete button. (In this case I've clicked on Mahinda Rajaraksa Delete button) | | |
| Expected Output | Delete record from the database. | Actual Output | Delete record from the database and from the Staff Page table. |
| Result | Success | | |
| Screenshot |  | | |

Table 4. 22 Test Case 21

| Test Case 21 | | | |
|-------------------------|---|----------------------|--------------------------|
| Function | Staff Page :- Back button functionality. | | |
| Requirement Type | System Requirement. | | |
| Inputs | In Staff Page click on back button. | | |
| Expected Output | Direct to the Home Page. | Actual Output | Direct to the Home Page. |
| Result | Success | | |
| Screenshot |  | | |

Table 4. 23 Test Case 22

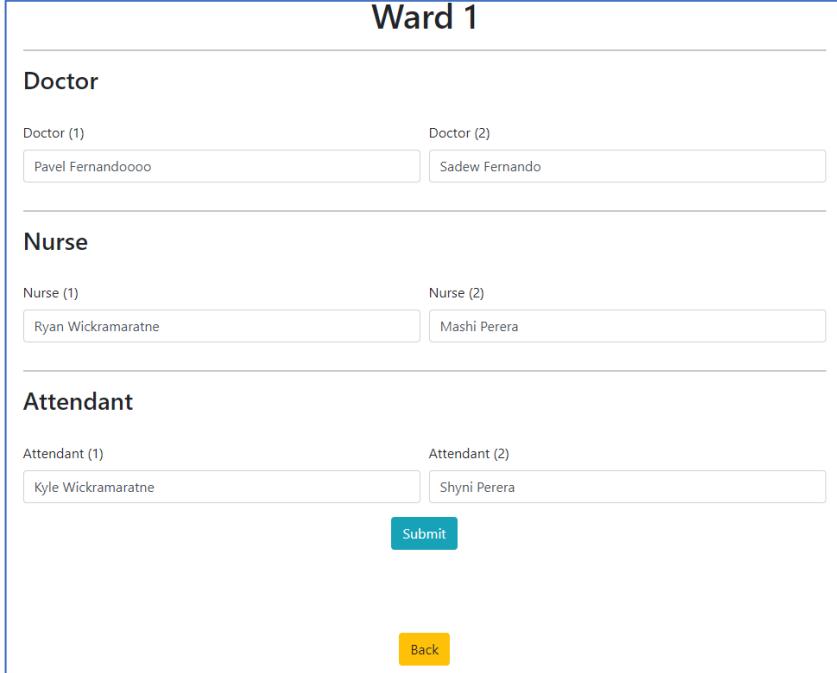
| Test Case 22 | | | |
|-------------------------|--|----------------------|----------------------------|
| Function | Ward Page :- Manage the ward button functionality. | | |
| Requirement Type | System Requirement. | | |
| Inputs | In Wards Page click on Manage the ward button (In this case I've clicked on Ward 1 row's Manage the Ward button. | | |
| Expected Output | Direct to the Ward 1 Page. | Actual Output | Direct to the Ward 1 Page. |
| Result | Success | | |
| Screenshot |  | | |

Table 4. 24 Test Case 23

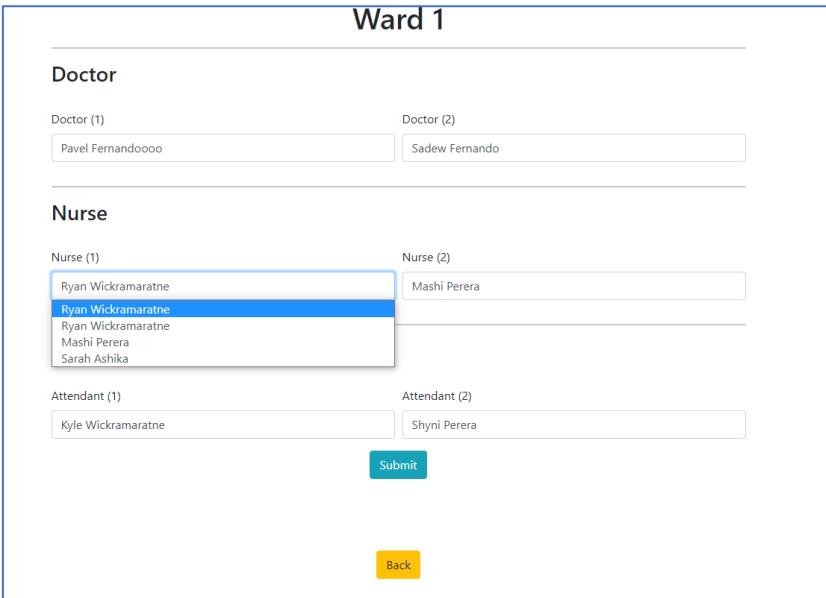
| Test Case 23 | | | |
|-------------------------|---|----------------------|------------------------------------|
| Function | Ward Page :- Retrieve data from database to combo box. | | |
| Requirement Type | System Requirement. | | |
| Inputs | In Ward1 Page click on click on any combo box. | | |
| Expected Output | Show relevant type of staff names. | Actual Output | Show relevant type of staff names. |
| Result | Success | | |
| Screenshot |  <p>The screenshot shows a user interface titled "Ward 1". It has sections for "Doctor" and "Nurse". Under "Nurse", there are two dropdown menus labeled "Nurse (1)" and "Nurse (2)". The "Nurse (1)" dropdown is open, showing a list of names: Ryan Wickramaratne, Ryan Wickramaratne, Mashi Perera, Sarah Ashika. The name "Ryan Wickramaratne" is highlighted with a blue selection bar. The "Nurse (2)" dropdown is empty, showing only the placeholder text "Mashi Perera". Below the dropdowns are sections for "Attendant (1)" and "Attendant (2)", each with a single entry: "Kyle Wickramaratne" and "Shyni Perera" respectively. At the bottom are "Submit" and "Back" buttons.</p> | | |

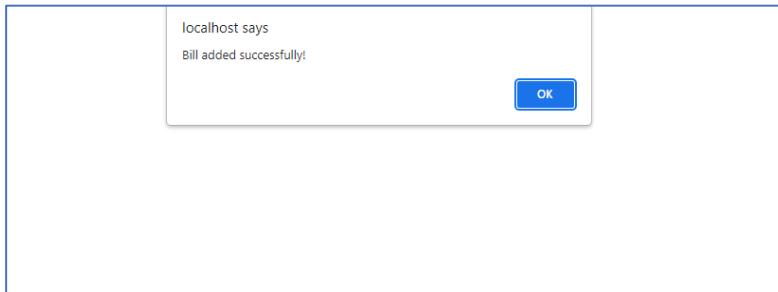
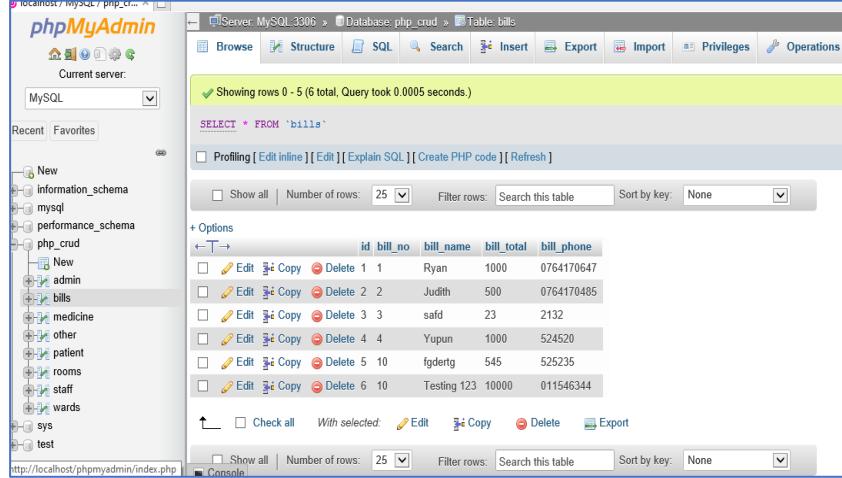
Table 4. 25 Test Case 24

| Test Case 24 | | | |
|-------------------------|--|----------------------|----------------------------------|
| Function | Ward Page :- Manage the Ward Room button functionality. | | |
| Requirement Type | System Requirement. | | |
| Inputs | In Rooms Page click on Manage the ward rooms button (In this case I've clicked on Ward 1 row's Manage the Ward Rooms button. | | |
| Expected Output | Direct to the Ward 1 Rooms Page. | Actual Output | Direct to the Ward 1 Rooms Page. |
| Result | Success | | |
| Screenshot |  | | |

Table 4. 26 Test Case 25

| Test Case 25 | | | |
|-------------------------|---|----------------------|----------------------------|
| Function | Room Page :- Logout button functionality. | | |
| Requirement Type | User Requirement. | | |
| Inputs | In any page of management click the red Logout button. | | |
| Expected Output | Direct back to login page. | Actual Output | Direct back to login page. |
| Result | Success | | |
| Screenshot |  | | |

Table 4. 27 Test Case 26

| Test Case 26 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------------------------|--|----------------------|---|------------|------------|---------|-----------|------------|------------|---|---|------|------|------------|--|---|---|--------|-----|------------|--|---|---|------|----|------|--|---|---|-------|------|--------|--|---|----|--------|-----|--------|--|---|----|-------------|-------|-----------|--|
| Function | Patient Page :- Patient Invoice data insert. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Requirement Type | User Requirement. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Inputs | In patient page click Patient Invoice button. Then fill the text boxes and click Enter Bill Button. In this case I've used Testing 123 as patient name. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Expected Output | Database with containing with patient invoice details. | Actual Output | Database with containing with patient invoice details. And notification with stating “Bill Added Successfully” | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Result | Success | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Screenshot |   <p>The screenshot shows the phpMyAdmin interface for a MySQL database named 'php_crud'. The 'bills' table is selected, displaying 6 rows of data:</p> <table border="1"> <thead> <tr> <th></th> <th>id</th> <th>bill_no</th> <th>bill_name</th> <th>bill_total</th> <th>bill_phone</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>Ryan</td> <td>1000</td> <td>0764170647</td> <td></td> </tr> <tr> <td>2</td> <td>2</td> <td>Judith</td> <td>500</td> <td>0764170485</td> <td></td> </tr> <tr> <td>3</td> <td>3</td> <td>safd</td> <td>23</td> <td>2132</td> <td></td> </tr> <tr> <td>4</td> <td>4</td> <td>Yupun</td> <td>1000</td> <td>524520</td> <td></td> </tr> <tr> <td>5</td> <td>10</td> <td>fgderg</td> <td>545</td> <td>525235</td> <td></td> </tr> <tr> <td>6</td> <td>10</td> <td>Testing 123</td> <td>10000</td> <td>011546344</td> <td></td> </tr> </tbody> </table> | | | | id | bill_no | bill_name | bill_total | bill_phone | 1 | 1 | Ryan | 1000 | 0764170647 | | 2 | 2 | Judith | 500 | 0764170485 | | 3 | 3 | safd | 23 | 2132 | | 4 | 4 | Yupun | 1000 | 524520 | | 5 | 10 | fgderg | 545 | 525235 | | 6 | 10 | Testing 123 | 10000 | 011546344 | |
| | id | bill_no | bill_name | bill_total | bill_phone | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | Ryan | 1000 | 0764170647 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 2 | Judith | 500 | 0764170485 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | 3 | safd | 23 | 2132 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | 4 | Yupun | 1000 | 524520 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | 10 | fgderg | 545 | 525235 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | 10 | Testing 123 | 10000 | 011546344 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

4.6 Evaluation results of the Test Plan

Table 4. 28 Evaluation results of the Test Plan

| Test Case No | Test Case | Result | What's need to be done more. |
|--------------|--|---------|---|
| 1 | Login Page:- Hide the password characters. | Success | * The login system should provide Two-Factor Authentication (2FA). It operates by adding an extra layer of security to accounts. It necessitates an additional login credential in addition to the username and password. |
| 2 | Login Page:- Prevent unauthorized users to login to the system. | Success | |
| 3 | Login Page :- Allowing admin to access to home page by giving correct login credentials. | Success | * Add forget password method to recover account if login credentials are forgotten. * Create a user sign up form :- Should create a new user sign up form is helping to register new admin or a user for system. Rather than depending on system developer for security purpose, this helps to enhance the security of the system further. |
| 4 | Home Page :- Submitting email to database by filling Sign Up for newsletter button. | Failed | * Fix the Sign Up for newsletter button. * Create a new table for add those data. |

| | | | |
|----------|--|---------|--|
| | | | <ul style="list-style-type: none"> * Create a dashboard :- Creating a dashboard for the system is needed. The dashboard displays all data from across the organization in one place. A unique dashboard provides important insights into the entire business, allowing users to see and analyze data. |
| 5 | Home Page :- Navigation Bar functionality. | Success | <ul style="list-style-type: none"> * Should give a separate navbar at the left side to make the navigation easier. * Insert a search bar :- Should create search bar for each form of the system helps to find recorded data in the database. |
| 6 | Website responsiveness. | Success | |
| 7 | About Us Page :- Getting lecturers' social media usernames or contact details. | Failed | <ul style="list-style-type: none"> * Create a new table to add lecturer's social media account details. * Add functionality to view lecturer's social media account details when clicking the relative social media platform icons. |
| 8 | Footer's positioning arrow icon functionality. | Success | |

| | | | |
|-----------|---|---------|---|
| 9 | Blog Page :- Read more button functionality. | Failed | * Add articles for relative blogs by adding more web pages for them. |
| 10 | Contact Us Page :- Accordion functionality. | Success | * Add google map functionality to get directions to the location easily. |
| 11 | Contact Us Page :- Map box functionality. | Success | |
| 12 | Staff Page :- Create New User button functionality. | Success | |
| 13 | Staff Page :- Insert New user record to database. | Success | * Should give ability to insert videos and photos when require (For example staff individuals photos). * Should enhance the form. For example, when user is submitted records to database the text boxes values should be automatically clear. |
| 14 | Add Staff Details Page :- Popup notification message when submitting records into database. | Failed | * Fix the popup notification message. |
| 15 | Add Staff Details Page :- Checking for validation. | Failed | * Provide a validation function to ensure that all mandatory fields are filled out. |
| 16 | Add Staff Details Page :- Data characters insert limitation. | Success | * Give relative error message when user entered long characters. |

| | | | |
|-----------|---|---------|---|
| 17 | Staff Page :- View button functionality. | Success | * Should give ability to printout the records. * Should give ability to export the records. |
| 18 | Staff Page :- Edit button functionality. | Success | * Should enhance the form. For example, when user is submitted records to database the text boxes values should be automatically clear. |
| 19 | Staff Page :- Edit and insert user record to database. | Success | * Add popup notification message when user edited records successfully. |
| 20 | Staff Page :- Delete user record from database. | Success | * Add popup notification message when user deleted records successfully. |
| 21 | Staff Page :- Back button functionality. | Success | |
| 22 | Ward Page :- Manage the ward button functionality. | Success | |
| 23 | Ward Page :- Retrieve data from database to combo box. | Success | |
| 24 | Ward Page :- Manage the Ward Room button functionality. | Success | |

| | | | |
|-----------|--|---------|---|
| 25 | Room Page :- Logout button functionality. | Success | * Should add the logout button in the nav bar. |
| 26 | Patient Page :- Patient Invoice data insert. | Success | * Should give ability to printout the invoices. |

4.7 Types of Software Testing

4.7.1 Introduction

The process of running a software to identify problems is known as testing. Our program must be error-free in order to work well. If the testing is successful, all errors in the software will be removed.

Testing Principles :

- All tests must meet the customer's specifications.
- A third party should undertake software testing for us.
- It is not possible to do thorough testing. Because we minimize the amount of testing possible depending on the application's risk assessment.
- Before putting it into action, all tests should be scheduled.
- It follows to the Pareto rule (the 80/20 rule), which claims that 80 percent of errors are caused by 20 percent of program components.
- Begin with little parts and work up to larger ones.

4.7.2 Testing Types

1) Unit Testing

It is concerned with the smallest unit of software design. In this test, we put an individual unit or a set of interconnected units to the test. The programmer frequently does this by using sample input and watching the related outputs.

Example:

- a) In a program, we check to see if the loop, method, or function is working properly.
- b) Incorrect or misunderstood arithmetic precedence.
- b) Improper initialization

2) Integration Testing

The goal is to use unit-tested components and construct a design-driven program framework. Integration testing is the process of combining a number of components to provide results.

Example:

- (a) Black Box testing:- It serves as validation. In this case, we ignore internal functioning mechanisms and concentrate on the result.
- (b) White box testing:- It serves as verification. This focuses on internal mechanics, or how the outcome is achieved.

3) Regression Testing

Every time a new module is added, the software changes. This form of testing ensures that the entire component functions properly even after it has been included to the overall program.

Example:

In school, suppose we have a module staff, students, and finance merging these modules and examining if the integration of these modules works properly in regression testing.

4) Smoke Testing

This test ensures that the program being tested is ready or safe for further testing. The testing of an initial pass is called a smoke test since it is done to see if it did not catch fire or smoke at the initial switch on.

Example:

If the project contains two modules, ensure that module one functions well before proceeding to module two.

5) Alpha Testing

This form of testing is known as validation testing. It is a type of acceptability testing performed prior to the product being distributed to customers. QA personnel are usually in charge of this.

Example:

When software testing is carried out within an organization

6) Beta Testing

The beta test is carried out at one or more customer sites by the software's end-user. This version is only available to a small number of people for testing in a real-time environment.

Example:

When software testing is done for a small group of people

7) System Testing

This software has been thoroughly tested to ensure that it is compatible with a variety of operating systems. It is covered under the black box testing method. We only focus on the required input and output without consideration for internal operations. Security testing, recovery testing, stress testing, and performance testing are all included.

Example:

This involves both functional and nonfunctional testing.

8) Stress Testing

We give the system unfavorable conditions and see how it performs in those conditions.

Example:

- (a) Test cases requiring the most memory or other resources are run.
- (b) In a virtual operating system, test cases that may induce thrashing.
- (c) Test scenarios that may require an excessive amount of storage space.

9) Performance Testing

It is intended to evaluate software run-time performance within the context of an integrated system. It is used to evaluate the program's speed and efficacy. It is also known as load testing. It determines the system's performance under a certain load.

Example:

Several processing cycles are being tested.

10) Object Oriented Testing

This testing is a combination of many testing methodologies that helps in the verification and validation of object-oriented software. This testing is carried out as follows: Requirements testing, testing design and analysis, testing code, integration testing, system testing, and user testing

11) Acceptance Testing

Customers do acceptance testing to determine whether or not the delivered items execute the tasks described in the specifications.

(Types of Software Testing, 2022)

4.8 Feedback Form

4.8.1 Feedback form of sample survey

The screenshot shows a Google Forms survey titled "USER FEEDBACK FORM". The survey instructions ask users to evaluate the system of Polly Pipe in Braintree and provides an email for contact. The survey consists of five questions:

- Write your name ?** (Text input field)
- What is your department ?** (Text input field)
- It is easy to log into the system ?** (Radio button choices: YES, NO)
- How do you rate the speed of the system ?** (Scale from 1 to 5, with 1 labeled "Very Slow" and 5 labeled "Vert Fast")
- How do you rate the security of the system ?** (Scale from 1 to 5, with 1 labeled "Very Unsecure" and 5 labeled "Very Secure")

Figure 4. 12 User feedback form part 1

USER FEEDBACK FORM

docs.google.com/forms/d/e/1FAIpQLSfyLM6Kr97SNcMYBDhTwrt2tp-mW9D-qZTqxdGKfW1OQxpQ...

How do you rate the security of the system ?

1 2 3 4 5
Very Unsecure ○ ○ ○ ○ ○ Very Secure

Is it easy to access authorized forms ?

Yes
 No
 Sometimes having errors

Does this system design help you to increase work efficiency ?

Yes
 No

If above answer is "NO", what is the reason ?

Your answer

How satisfied are you with our product ?

1 2 3 4 5
Very Unsatisfied ○ ○ ○ ○ ○ Very Satisfied

How satisfied were you with the instructions before use ?

1 2 3 4 5

Figure 4. 13 User feedback form part 2

USER FEEDBACK FORM

How satisfied were you with the instructions before use ?

1 2 3 4 5

Very Unsatisfied Very Satisfied

How satisfied is the quality of the product ?

1 2 3 4 5

Very Unsatisfied Very Satisfied

How easy is it to find data from the database

1 2 3 4 5

Very Hard Very Easy

How would you rate the design of the system ?

1 2 3 4 5

Very Unsatisfied Very Satisfied

How satisfied are you with our service ?

1 2 3 4 5

Very Unsatisfied Very Satisfied

How satisfied are you with our team?

!  

Figure 4. 14 User feedback form part 3

USER FEEDBACK FORM

How would you rate the design of the system?

1 2 3 4 5

Very Unsatisfied Very Satisfied

How satisfied are you with our service ?

1 2 3 4 5

Very Unsatisfied Very Satisfied

How satisfied are you with our team?

1 2 3 4 5

Very Unsatisfied Very Satisfied

Would you like to a representative to contact you ?

Yes
 No

Tell us how we can improve ?

Your answer

Submit **Clear form**

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

Google Forms

Figure 4. 15 User feedback form part 4

4.8.2 How I can improve the feedback collection form

Customer feedback is crucial since it acts as a direction for the company's future development. Companies are always interested in learning what they're doing correctly and wrong in the perspective of their customers. Positive solutions can be found in both the good and the poor, making it easier to change and adapt the client experience over time. In a brief, feedback is how we ensure that our community is at the center of everything we do.

1) Using simple images for customer rating scores

When rating a characteristic of something it is easy to rate it by selecting simple images rather than reading a numeral score. And it is very attractive way of keep customer adhere to the feedback form with an interest.

How much effort did you personally have to put forth to handle your request?

The figure shows a horizontal scale for rating effort. It consists of five circular icons arranged horizontally. From left to right, the icons represent increasing levels of effort: a sad face, a slightly frowny face, a neutral face, a smiling face, and a very happy face. Below the first icon is the text "Very high effort". Below the last icon is the text "Very low effort".

Figure 4. 16 Using simple images for customer rating scores

2) Making a quick survey

However, people are more likely to complete our survey if it is brief. That doesn't mean we have to compromise on the survey's quality. All we have to do now is ask the correct questions and ask for a quick response. That demanding response should have the potential to lead to a great, in-depth response.

3) Aim for a beautiful design that is also simple to use.

On whatever device, the feedback form should be professional, tidy, and easy to navigate. Consider including a corporate logo and using company colors to make it more attractive. A basic, tasteful design will always provide an excellent first impression, and it makes the survey more interesting to complete.

4) Side feedback button at the edge of the system

We can adjust the side button, which sits at the system's edge, waiting for a consumer to click it, which will launch the feedback form and prompt them to take action.

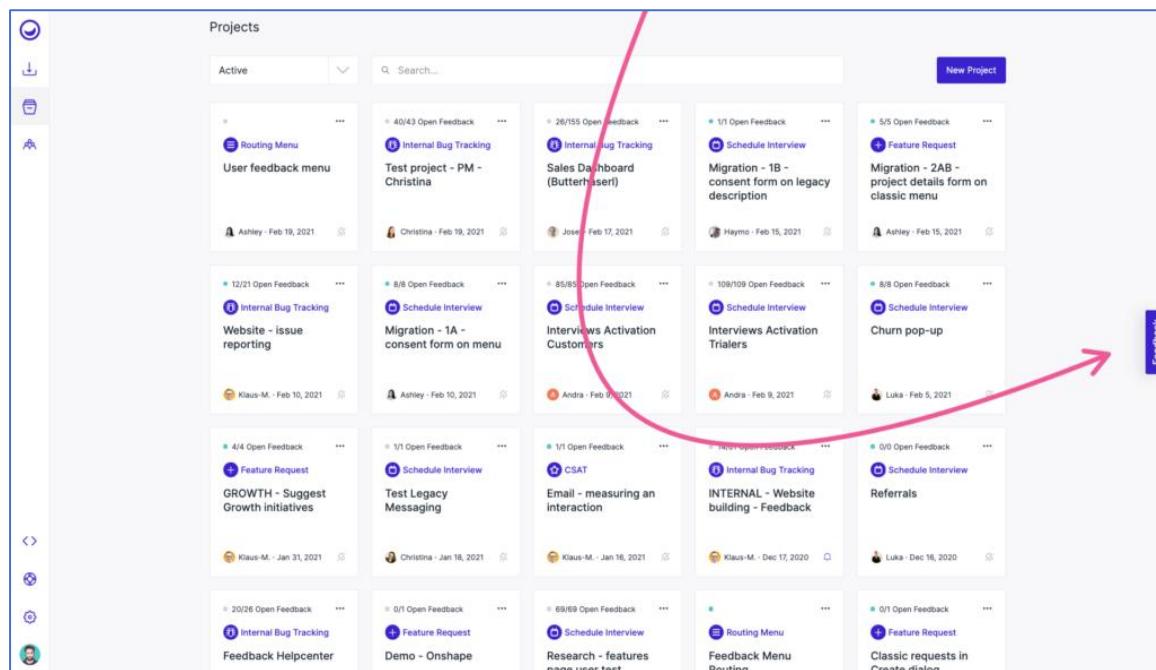


Figure 4. 17 Side feedback button at the edge of the system

5) Instant feedback from cooperate website

We can get fast consumer feedback without asking any questions using an embeddable on-site widget.

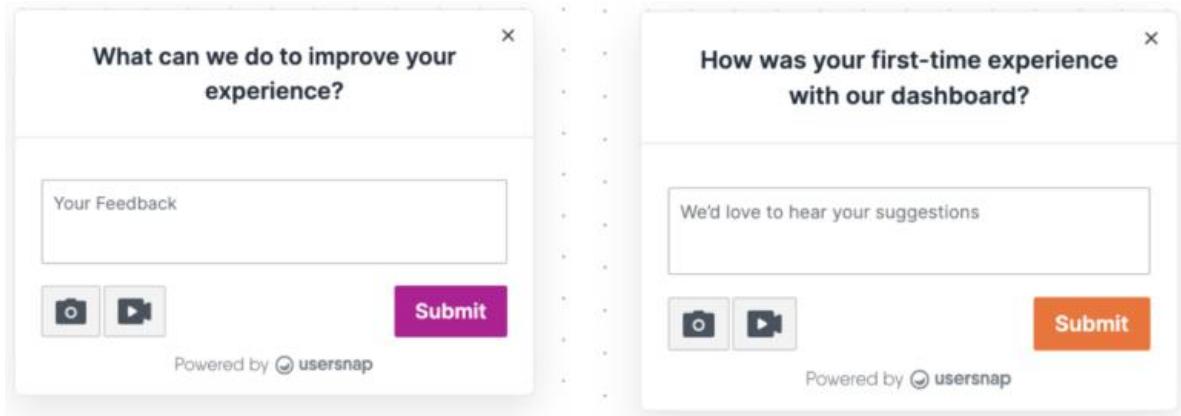


Figure 4. 18 Instant feedback sample from cooperate website

4.9 Future Enhancements

Managing software quality has become an essential component of project management at all levels. Delivering projects with high-quality software will be more cost-effective and efficient. Finding ways to adopt effective testing strategies as soon as possible will aid in the detection and resolution of issues.

- Make a more secured cloud-based database :- Moving the database to the cloud, as cloud services develop and become more economical, can provide flexible, affordable, and scalable database management. A reliable and efficient database connection is required for Windows applications. A functional, flexible, and secure database serves as the foundation for developing interesting application.
- Create a dashboard :- Creating a dashboard for the system is needed. The dashboard displays all data from across the organization in one place. A unique dashboard provides important insights into the entire business, allowing users to see and analyze data.
- Create a user sign up form :- Should create a new user sign up form is helping to register new admin or a user for system. Rather than depending on system developer for security purpose, this helps to enhance the security of the system further.
- Insert a search bar :- Should create search bar for each form of the system helps to find recorded data in the database.
- Should give ability to printout the records.
- Should give ability to printout the invoices.
- Should give ability to export the records.
- Should give ability to insert videos and photos when require (For example staff individuals photos).
- For some pages should insert popup buttons to make more attractive and to increase user experience.
- Should use multi-factor authentication to secure accounts.
- Should give forget password functionality which helps users who have forgotten their password can unlock, recover, or reset it by answering account secret questions or sending an e-mail.

- Should enhance the Create/Update forms. For example, when user is submitted records to database the text boxes values should be automatically clear.
- Should add the logout button in the nav bar.
- Should give a separate navbar at the left side to make the navigation easier.

Conclusion

This entire assignment is based on an implementation of a fully functional system installation for Arogya Hospital. The purpose of this assignment is to improve web design and developing skills.

This report includes details about communication protocols, details about web servers and types of servers relate to web development. And details about DNS, SEO and a documentation about types of web tools and technologies.

And also, this assignment gives much more details about front end development, back-end development and full stack with describing 3 layer architecture. And have compared online web creation tools and customs tools.

And also have basic database software handling, creating designs, understanding data flows, understanding different database designing methods, working with IDEs, Basic system creation using Microsoft SQL and WAMP Server and other many more technologies such as bootstrap and etc.

And used different SQL query types when creating a database for a system, make a documentation for a system with describing future enhancements for the system.

References

CareerFoundry. 2022. The 7 Most Essential Frontend Web Development Tools in 2022.

[ONLINE] Available at: <https://careerfoundry.com/en/blog/web-development/7-essential-tools-for-front-end-development/>.

[Accessed 07 June 2022].

Sammi Merritt. 2022. The Benefits of Using Web Design Templates | Atilus.

[ONLINE] Available at: <https://www.atilus.com/benefits-using-web-design-templates/>.

[Accessed 07 June 2022].

Icecube Digital. 2022. Advantages of Custom Built Sites vs Online Website Builder Tools.

[ONLINE] Available at: <https://www.icecubedigital.com/blog/advantages-of-custom-built-sites-vs-online-website-builder-tools/>.

[Accessed 07 June 2022].

Shelley Johansson. 2022. The difference between template and custom-built websites.

[ONLINE] Available at: <https://www.primedesignsolutions.com/learning-center/difference-template-custom-built-websites/>.

[Accessed 07 June 2022].

StacyB. 2022. Custom Web Design vs. Website Template - MayeCreate DesignMayeCreate Design.

[ONLINE] Available at: <https://mayecreate.com/blog/custom-web-design-vs-website-template/>.

[Accessed 10 June 2022].

WEBO Digital. 2022. "Custom-Built" Vs "Template" Websites - What's Important & Which Is Better? - WEBO Digital.

[ONLINE] Available at: <https://webo.digital/blog/custom-built-vs-template-websites-which-is-better/>.

[Accessed 10 June 2022].

SumatoSoft. 2022. Why use Ruby on Rails for Your Software Development Project.

[ONLINE] Available at: <https://sumatosoft.com/blog/why-use-ruby-on-rails-for-your-software-development-project>.

[Accessed 10 June 2022].

Guru99. 2022. What is Backend Developer? Skills Need for Web Development.

[ONLINE] Available at: <https://www.guru99.com/what-is-backend-developer.html>.

[Accessed 10 June 2022].

Difference Between Frontend and Backend | Difference Between. 2022. Difference Between Frontend and Backend | Difference Between.

[ONLINE] Available at: <http://www.differencebetween.net/technology/difference-between-frontend-and-backend/>.

[Accessed 10 June 2022].

GeeksforGeeks. 2022. Frontend vs Backend - GeeksforGeeks.

[ONLINE] Available at: <https://www.geeksforgeeks.org/frontend-vs-backend/>.

[Accessed 15 June 2022].

UofT SCS Boot Camps. 2022. What Is a Full Stack Developer & What Do They Do? - UofT SCS Boot Camps.

[ONLINE] Available at: <https://bootcamp.learn.utoronto.ca/blog/what-is-a-full-stack-developer/#:~:text=What%20is%20a%20Full%20Stack%20Engineer%3F,can%20see%20and%20interact%20with..>

[Accessed 15 June 2022].

What Is the Difference Between Front-End and Back-End Development?. 2022. What Is the Difference Between Front-End and Back-End Development?.

[ONLINE] Available at: <https://www.conceptatech.com/blog/difference-front-end-back-end-development>.

[Accessed 15 June 2022].

Learning Center. 2022. What is OSI Model | 7 Layers Explained | Imperva.

[ONLINE] Available at: <https://www.imperva.com/learn/application-security/osi-model/>.

[Accessed 15 June 2022].

Web Applications | Arcitura Patterns. 2022. Web Applications | Arcitura Patterns.

[ONLINE] Available at: https://patterns.arcitura.com/cloud-computing-patterns/basics/web-technology/web_applications.

[Accessed 15 June 2022].

What is Three-Tier Architecture | IBM. 2022. What is Three-Tier Architecture | IBM.

[ONLINE] Available at: <https://www.ibm.com/cloud/learn/three-tier-architecture#:~:text=Three%2Dtier%20architecture%20is%20a,associated%20with%20the%20application%20is.>

[Accessed 20 June 2022].

How Improving Page Speed Boosts SEO Rankings | Pantheon. 2022. How Improving Page Speed Boosts SEO Rankings | Pantheon.

[ONLINE] Available at: <https://pantheon.io/blog/improving-page-speed-boosts-seo-ranking#:~:text=Keep%20in%20mind%20that%20the,stacks%20up%20against%20the%20competition..>

[Accessed 20 June 2022].

Faraz Ali. 2022. How SEO Will Affect Your Website Performance | by Faraz Ali | Medium.

[ONLINE] Available at: <https://medium.com/@farazali25524/how-seo-will-affect-your-website-performance-6bba7219a75b>.

[Accessed 20 June 2022].

Optimizely. 2022. Search engine optimization - Optimizely .

[ONLINE] Available at: <https://www.optimizely.com/optimization-glossary/search-engine-optimization/>.

[Accessed 20 June 2022].

SearchDataCenter. 2022. Learn the major types of server hardware and their pros and cons.

[ONLINE] Available at: <https://www.techtarget.com/searchdatacenter/feature/Learn-the-major-types-of-server-hardware-and-their-pros-and-cons>.

[Accessed 20 June 2022].

WhatIs.com. 2022. What is a Server? - Definition from WhatIs.com.

[ONLINE] Available at: <https://www.techtarget.com/whatis/definition/server>.

[Accessed 20 June 2022].

The Difference Between Windows Server And Windows | Avast. 2022. The Difference Between Windows Server And Windows | Avast.

[ONLINE] Available at: <https://blog.avast.com/difference-between-windows-server-and-windows-avast>.

[Accessed 25 June 2022].

WhatsaByte. 2022. Windows 10 vs Windows Server: [Differences Explained] | WhatsaByte.

[ONLINE] Available at: <https://whatsabyte.com/windows-10-vs-windows-server#:~:text=But%20what%20are%20the%20main,meant%20for%20running%20server%20systems..>

[Accessed 25 June 2022].

How-To Geek. 2022. What's the Difference Between Windows and Windows Server? .

[ONLINE] Available at: <https://www.howtogeek.com/404763/whats-the-difference-between-windows-and-windows-server#:~:text=Windows%2010%20and%20Windows%20Server%20Share%20Similar%20Code&text=They%20can%20have%20the%20same,or%20Microsoft%20Office%20on%20both..>

[Accessed 25 June 2022].

We The Geek. 2022. Difference Between Windows & Windows Server.

[ONLINE] Available at: <https://wethegeek.com/difference-between-windows-windows-server/>.

[Accessed 25 June 2022].

Pittsburgh. 2022. Network Server Types Explained .

[ONLINE] Available at: <https://www.nhpittsburgh.com/solutions/resources/upcoming-events-and-webinars/network-server-types-explained>.

[Accessed 25 June 2022].

www.youtube.com. 2022. No page title.

[ONLINE] Available at: https://www.youtube.com/watch?v=_hdtdJWJGtA.

[Accessed 25 June 2022].

YouTube. 2022. Comparing Windows 10 to Windows Server 2016 - YouTube.

[ONLINE] Available at: <https://www.youtube.com/watch?v=L8hfFUGUBfo>.

[Accessed 25 June 2022].

We The Geek. 2022. Difference Between Windows & Windows Server.

[ONLINE] Available at: <https://wethegeek.com/difference-between-windows-windows-server/>.

[Accessed 25 June 2022].

Apachebooster Blog: Showcasing the tech blogs written by our writers. 2022. What is server clustering and how does it work?.

[ONLINE] Available at: <https://apachebooster.com/blog/what-is-server-clustering-and-how-does-it-work/>.

[Accessed 25 June 2022].

www.javatpoint.com. 2022. Server Operating System - javatpoint.

[ONLINE] Available at: <https://www.javatpoint.com/server-operating-system>.

[Accessed 30 June 2022].

Shortie Designs. 2022. 10 Top Principles of Effective Web Design - Shortie Designs.

[ONLINE] Available at: <https://shortiedesigns.com/blog/10-top-principles-effective-web-design/>.

[Accessed 30 June 2022].

Lucidchart. 2022. UML Class Diagram Tutorial | Lucidchart.

[ONLINE] Available at: <https://www.lucidchart.com/pages/uml-class-diagram>.

[Accessed 30 June 2022].

F5 Studio. 2022. What Is Quality Assurance And Why QA Is Important in Web Development.

[ONLINE] Available at: <https://f5-studio.com/articles/what-is-qa-and-why-quality-assurance-is-important-in-web-development-process/>.

[Accessed 30 June 2022].

Synoptek. 2022. Types of QA Testing: Everything you Need to Know | Synoptek.

[ONLINE] Available at: <https://synoptek.com/insights/it-blogs/types-of-qa-testing-everything-you-need-to-know/>.

[Accessed 30 June 2022].

Unleashed Technologies. 2022. Website Quality Assurance | Unleashed.

[ONLINE] Available at: <https://www.unleashed-technologies.com/blog/website-quality-assurance>.

[Accessed 30 June 2022].

SEOptimer: SEO Audit & Reporting Tool. White Label Option.. 2022. Your Complete Guide to Website QA (Quality Assurance) with Free QA Checklist.

[ONLINE] Available at: [https://www.seoptimer.com/blog/website-qa/#:~:text=Website%20QA%20\(Quality%20Assurance\)%20can,earlier%2C%20even%20before%20development%20begins..](https://www.seoptimer.com/blog/website-qa/#:~:text=Website%20QA%20(Quality%20Assurance)%20can,earlier%2C%20even%20before%20development%20begins..)

[Accessed 30 June 2022].

Software Testing and Quality Assurance by iBeta. 2022. Test Plan Versus Test Case; What's the Difference? – Software Testing and Quality Assurance by iBeta.

[ONLINE] Available at: <https://www.ibeta.com/test-plan-versus-test-case-whats-the-difference/#:~:text=A%20test%20plan%20is%20a,Goals%20and%20objectives.>

[Accessed 04 July 2022].

YouTube. 2022. How a DNS Server (Domain Name System) works. - YouTube.

[ONLINE] Available at:

https://www.youtube.com/watch?v=mpQZVYPuDGU&vl=en&ab_channel=PowerCertAnimatedVideos.

[Accessed 04 July 2022].

Liquid Web. 2022. Understanding the DNS Process | Liquid Web.

[ONLINE] Available at: https://www.liquidweb.com/kb/how-to-demystify-the-dns-process/?irgwc=1&clickid=QwyU9FU%3AyxyIT6m2yR1sJXLLUkGVaGXB7XFNUw0&iradid=278394&irpid=269814&sharedid=03ea2ee84c5164fcd9b27f5c9020a984&_ir=1&utm_medium=affil <https://www.cloudflare.com/learning/dns/what-is-dns/iate>.

[Accessed 10 July 2022].

Clifford Chi. 2022. The 5 Basic Parts of a URL: A Short Guide.

[ONLINE] Available at: <https://blog.hubspot.com/marketing/parts-url>.

[Accessed 10 July 2022].

Novell Documentation. 2022. Novell Documentation.

[ONLINE] Available at:

http://www.novell.com/documentation/dns_dhcp/?page=/documentation/dns_dhcp/dhcp_enu/data/behbhhj.html.

[Accessed 10 July 2022].

Pediaa.Com. 2022. What is the Difference Between Static and Dynamic Website - Pediaa.Com.

[ONLINE] Available at: <https://pediaa.com/what-is-the-difference-between-static-and-dynamic-website/>.

[Accessed 10 July 2022].

WhatIs.com. 2022. What is a Web Server and How Does it Work?.

[ONLINE] Available at: <https://www.techtarget.com/whatis/definition/Web-server>.

[Accessed 10 July 2022].

GeeksforGeeks. 2022. Web Server and Its Type - GeeksforGeeks.

[ONLINE] Available at: <https://www.geeksforgeeks.org/web-server-and-its-type/>.

[Accessed 10 July 2022].

YouTube. 2022. SSL, TLS, HTTP, HTTPS Explained - YouTube.

[ONLINE] Available at:

https://www.youtube.com/watch?v=hExRDVZHhig&t=12s&ab_channel=PowerCertAnimatedVideos.

[Accessed 10 July 2022].

YouTube. 2022. FTP (File Transfer Protocol), SFTP, TFTP Explained. - YouTube.

[ONLINE] Available at:

https://www.youtube.com/watch?v=tOj8MSEIbfA&t=1s&ab_channel=PowerCertAnimatedVideos.

[Accessed 10 July 2022].

Techopedia. 2022. What is a Communication Protocol? - Definition from Techopedia.

[ONLINE] Available at: <https://www.techopedia.com/definition/25705/communication-protocol>.

[Accessed 10 July 2022].