**MY DOCTORS**

**A PROJECT REPORT**

***Submitted by :***

1. **Puneet Sharma (21BCE10647)**
2. **Shiivam Kasaudhan (21BCE10670)**
3. **Prince Jaiswal (21BCE10649)**
4. **Kanishq Kadam (21BCE10676)**
5. **Jai Singh Bisht (21BCE10684)**

*in partial* fulfillment *for the award of the degree*  *of*

**BACHELOR OF TECHNOLOGY**

*in*

# PROGRAM OF STUDY



**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING**

**VIT BHOPAL UNIVERSITY**

**KOTHRIKALAN, SEHORE**

**MADHYA PRADESH - 466114**

# September 2022

**BONAFIDE CERTIFICATE**

Certified that this project report titled **“MYDOCTORS”** is the bonafide work of

**“Puneet Sharma (21BCE10647) ,ShiivamKasaudhan**

**(21BCE10670), Prince Jaiswal (21BCE10649) , Kanishq**

**Kadam(21BCE10676) , Jai Singh Bisht (21BCE10684)”** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported here does not form part of any other project / research work on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**PROGRAM CHAIR PROJECT GUIDE**

**Dr Pritam Suman Dr. Anand Motwani**

School of Computer Science and Engineering School of Computer Science and Engineering VIT BHOPAL UNIVERSITY VIT BHOPAL UNIVERSITY

The Project Exhibition I Examination is held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# ACKNOWLEDGEMENT

I wish to express my heartfelt gratitude to **Dr Pritam Suman**, Head of the Department of Computer Science for much of his valuable support encouragement in carrying out this work.

I would like to thank my internal guide **Dr. Anand Motwani**, for continually guiding and actively participating in my project, giving valuable suggestions to complete the project work.

I would like to thank all the technical and teaching staff of the School of Computer Science, who extended directly or

indirectly all support.

Last, but not the least, I am deeply indebted to my parents who have been the greatest support while I worked day and night for the project to make it a success.

## TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER**  **NUMBER** | **Title** | **Page No** |
|  | List of Abbreviations  Abstract | 6  8 |
| **1** | **INTRODUCTION**  1.1 Motivation for the work  1.2 The Problem  1.3 Introduction to project | 9 |
| **2** | **REQUIREMENT ANALYSIS**  2.1 Customer requirements  2.2 Data requirements  2.3 Software requirements | 12 |
| **3** | DEFINING 3.1 Introduction  3.2 Data Collection  3.3 Our Features | 15 |
| **4** | **SYSTEM DESIGNING**  4.1 Introduction  4.2 Designs used | 17 |
| **5** | **CODING AND IMPLEMENTATION**  5.1 Front End  5.2 Back End and database | 19 |
| **6** | **TESTING and DEPLOYMENT**  6.1 Usability testing  6.2 Functional testing  6.3 Compatibility testing  6.4 Database testing  6.5 DEPLOYMENT | 26 |
| **7** | **FUTURE ENHANCEMENT AND CONCLUSION**  7.1 Future Enhancements  7.2 Conclusion | 40 |
|  | **References** | 42 |
|  |  |  |

### LIST OF ABBREVIATIONS

**HTML**: HTML is an acronym for Hypertext Markup Language and is the standard language for documents that have been designed to be displayed in a web browser.

**CSS:** CSS is an acronym that stands for Cascading Style Sheets which is a style sheet language that is used to express how a structured document like an HTML document should look.

**JS:** JS is an abbreviation for JavaScript. It is a scripting language that is used to create and control dynamic content on a webpage.

**MYSQL:** MySQL Cluster enables users to meet the database challenges of next generation web, cloud, and communications services with uncompromising scalability, uptime and agility.

**PHP:**PHP code is usually processed on a [web server](https://en.wikipedia.org/wiki/Web_server) by a PHP [interpreter](https://en.wikipedia.org/wiki/Interpreter_(computing)) implemented as a [module,](https://en.wikipedia.org/wiki/Plugin_(computing)) a [daemon](https://en.wikipedia.org/wiki/Daemon_(computing)) or as a [Common Gateway Interface](https://en.wikipedia.org/wiki/Common_Gateway_Interface) (CGI) executable.

### ABSTRACT

Effective doctor-patient communication is a central clinical function in building a therapeutic doctor-patient relationship, which is the heart and art of medicine. This is important in the delivery of high-quality health care. Much patient dissatisfaction and many complaints are due to breakdown in the doctor-patient relationship. However, many doctors tend to overestimate their ability in communication. Over the years, much has been published in the literature on this important topic.

**MYDOCTORS** is an online webpage project that’s serves the functionality of searching best doctors around your locality. Our system allows only registered users to log in and new users are allowed to register on the web page, this project provides useful product to all the age categories who are in need of the consulting doctors, it allows the users to give overview of different doctors through different cities. Once the user selects city, they are able to view the list of experienced doctors.

All the data is then send to the administrator (website owner) and they may interact with the doctors and patients, We maintain data of both users and doctors in our database and we can provide best outcome to both entities.**MEESHO is an ecommerce website where people can buy from wide range of products and has an interactive UI.**

**CHAPTER 1**

**INTRODUCTION**

### Motivation for the work

Engineers have always believed that we have the potential to learn new technology and apply it to real-world issues in order to create solutions that can be implemented on a big scale. We were handed this project work by the university, and it sparked a passion within all of us. We did extensive research on the issue of the shortage of doctors and medical facilities in our nation for this project. While this circumstance grieved us, it also provided us with the incentive we needed to approach this project with greater openness.

Regular insights from our mentor helped us stay motivated and commit to our project with a more realistic level of commitment. The conviction that our project will be successfully implemented on a large scale also serves as motivation for our endeavour.

#### **1.2 The Problem**

There is a huge gap in our healthcare system of **less availability of doctors and healthcare facilities to common people**.

In India, on an average, a doctor attends to 11,082 people, more than 10 times than what the WHO Recommends. Over the last nine years, shortage of doctors claimed lives of 72,000 infants in government hospitals of Madhya Pradesh.

In Bihar, one government doctor serves 28,391 people. Uttar

Pradesh is ranked second with 19,962 patients per doctor,

which is followed by Jharkhand (18,518), Madhya Pradesh

(16,996), Chhattisgarh (15,916) and Karnataka (13,556).

Delhi is better in terms of doctor-population ratio (1:2203), but

it is still twice the ratio recommended by WHO. The states

and UTS that are closest to meeting the WHO standards are

Arunachal Pradesh, Puducherry, Manipur and Sikkim.

**Thus, this problem of not able to find doctors and medical facilities easily and quickly at times when it is needed is huge in magnitude.**

##### 1.3 Introduction to project (Solution)

Our "MY DOCTOR" initiative is a medical website. My doctors have made it possible for users (patients) to use our now flagship function, which makes it simple to find for doctors in particular geographic locations and specialties according to their needs. Additionally, we have given patients and users a way to contact us with any questions they may have regarding their health.

In order to help our users, lead healthy lifestyles, we have also provided them with informative blogs about health-related topics.

**CHAPTER 2**

**REQUIREMENTS ANALYSIS**

2.1 Customer requirements

* Easy and quick navigation from homepage to specific features for specific needs.
* Searching doctors for particular area or for specific specialization.
* Reaching out to us for heath related query.
* Highly-responsive mobile site

2.2 Data requirements

* A customer can login and sign up their details such as name, DOB and contact number. They can also view their details in user dash boards

2.3 Software requirements

Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

Operating system: windows 7/ xp/8

Front end: html, css, java script

Server-side script: php

Database: mysql

* Architecture diagram: An architecture diagram is a visual representation of all the elements that make up part, or all, of a system. Above all, it helps the engineers, designers, stakeholders — and anyone else involved in the project — understand a system or app’s layout.

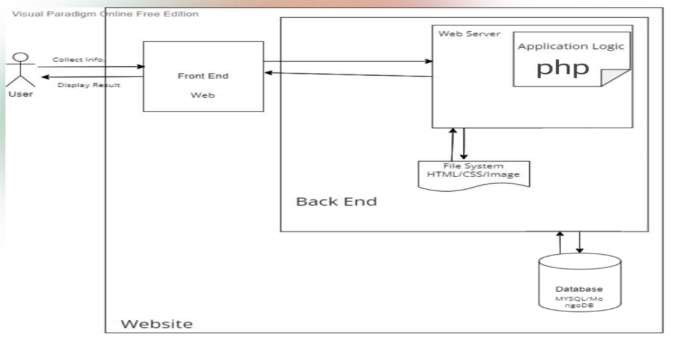


FIG.2.1

The architecture design for our project MYDOCTORS is shown in figure 4.1 above. By using this architecture diagram, we may better understand our system and get support for its development and implementation. The diagram gives us a clear concept of how the user (patient) on our platform will interact with our website utilizing an interactive frontend interface and then retrieve the needed doctor information from our MySQL built database as needed. Our backend, which we built primarily using PHP, will be used to facilitate this connection between our frontend and database. Our database has complete information about doctors, including their specialties, the location of their clinic or hospital, and their contact information. The database also contains patient data, such as their name, date of birth, and contact information.

* Use Case diagram: A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has

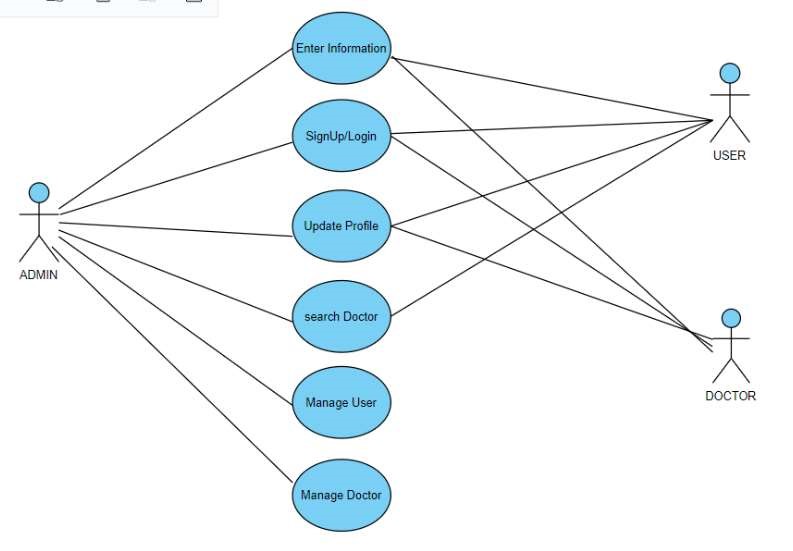


Fig 2.2

**CHAPTER 3**

**DEFINING**

3.1 Introduction:

Once the requirement analysis is done the next step is to clearly define and document the product features and get them approved from the customer or through market analysis.

We searched for different healthcare websites presently in market relating to our project domain and we did research on following combinations: interface design, features that are offered, usability and user interactivity. We also analysed monetization features and the loops in existing top 3-4 listed websites.

3.2 Data Collection:

We saw for data that would be the relevant for creating this website. Thus, we concluded that in this project the main data will be of doctors, their locations, contact details and specializations. Accordingly, we researched for material regarding this project and tried to implement it in our project database.

3.3 Our Features:

* Interactive Home page
* Easy Sign up and sign in
* Doctors search using location and specialization.
* Health related query.
* Social Links
* Doctor highlights
* Review
* Health blogs

### CHAPTER 4

**SYSTEM DESIGNING**

4.1 Introduction

This project “My Doctor” is based on the web development wherein we have used languages like HTML, CSS to build the front-end development part of our platform. The back-end uses the mainstream work of PHP and for creating the system database we have used MySQL using WampServer.

4.2 Designs used

For designing system diagrams, we have used visual paradigm software.

* Class diagram: a class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects.

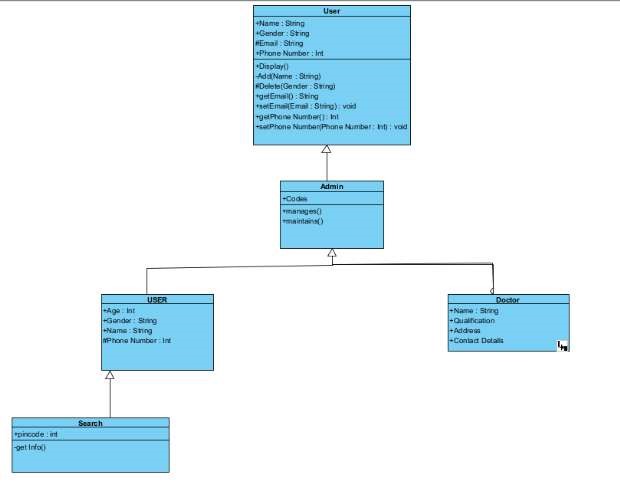


Fig 4.1

**CHAPTER 5**

**CODING AND IMPLEMENTATION**

#### 5.1 Front end

* HTML: Language used in creating structure of web pages.

HTML-The Hypertext Markup Language or HTML is the standardmarkup languag[e](https://en.wikipedia.org/wiki/Markup_language) for documents designed to be displayed in a [web browser.](https://en.wikipedia.org/wiki/Web_browser) It can be assisted by technologies such as [Cascading Style Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets) (CSS) and [scripting languages](https://en.wikipedia.org/wiki/Scripting_language) such as [JavaScript.](https://en.wikipedia.org/wiki/JavaScript)

[Web browsers](https://en.wikipedia.org/wiki/Web_browser) receive HTML documents from a [web server](https://en.wikipedia.org/wiki/Web_server) or from local storage and [render](https://en.wikipedia.org/wiki/Browser_engine) the documents into multimedia web pages. HTML describes the structure of a [web page](https://en.wikipedia.org/wiki/Web_page) [semantically](https://en.wikipedia.org/wiki/Semantic_Web) and originally included cues for the appearance of the document.

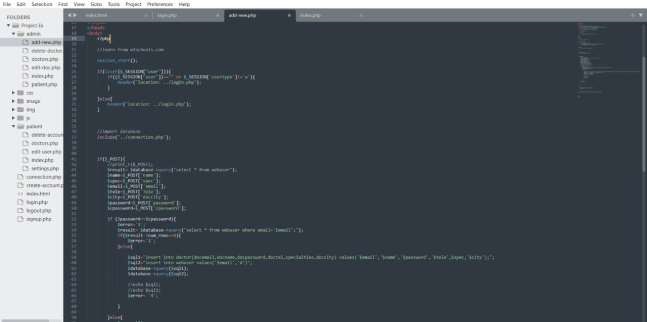
[HTML elements](https://en.wikipedia.org/wiki/HTML_element) are the building blocks of HTML pages. With HTML constructs, [images](https://en.wikipedia.org/wiki/HTML_element#Images_and_objects) and other objects such as [interactive forms](https://en.wikipedia.org/wiki/Fieldset) may be embedded into the rendered page. HTML provides a means to createstructured document[s](https://en.wikipedia.org/wiki/Structured_document) by denoting structural [semantics](https://en.wikipedia.org/wiki/Semantics) for text such as headings, paragraphs, lists, [links,](https://en.wikipedia.org/wiki/Hyperlink) quotes and other items.

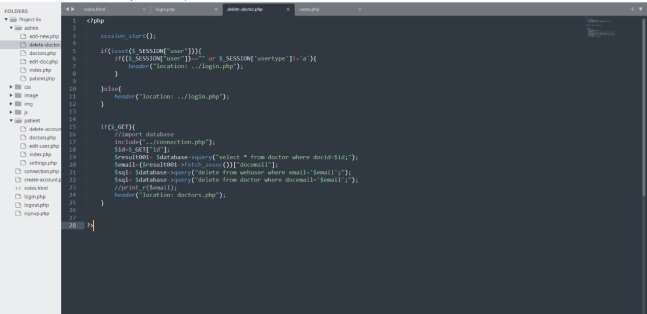
* CSS- CSS is designed to enable the separation of presentation and content, including [layout,](https://en.wikipedia.org/wiki/Page_layout) [colors,](https://en.wikipedia.org/wiki/Color) and [fonts.](https://en.wikipedia.org/wiki/Typeface) This separation can improve content [accessibility;](https://en.wikipedia.org/wiki/Accessibility) provide more flexibility and control in the specification of presentation characteristics; enable multiple [web pages](https://en.wikipedia.org/wiki/Web_page) to share formatting by specifying the relevant CSS in a separate css file, which reduces complexity and repetition in the structural content; and enable the css file to be [cached](https://en.wikipedia.org/wiki/Cache_(computing)) to improve the page load speed between the pages that share the file and its formatting.

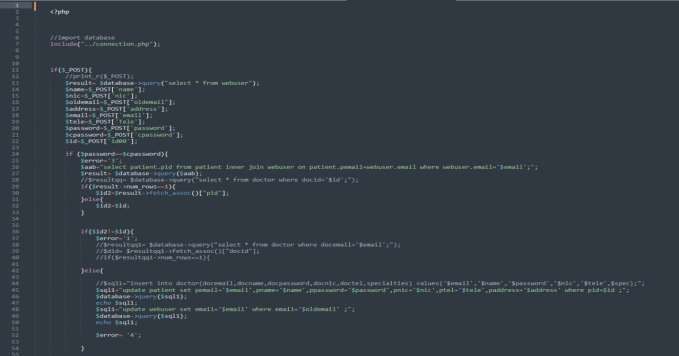
#### 5.2 Back End and database

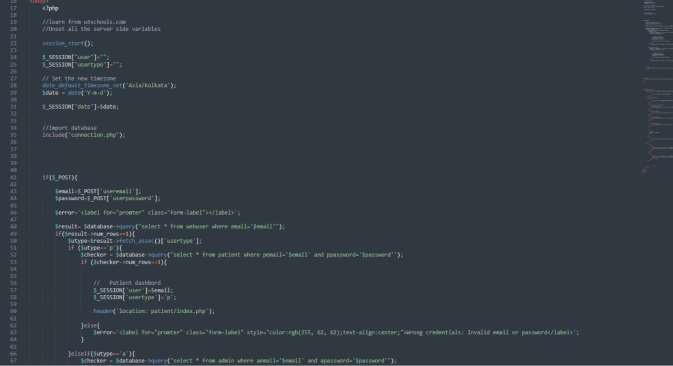
* PHP –PHP can actually do anything related to server-side scripting or more popularly known as the backend of a website. For example, PHP can receive data from forms, generate dynamic page content, can work with databases, create sessions, send and receive cookies, send emails etc. There are also many hash functions available in PHP to encrypt user’s data that makes PHP secure and reliable to be used as a server-side scripting language. So these are some of the abilities of PHP that makes it suitable to be used as server-side scripting language.

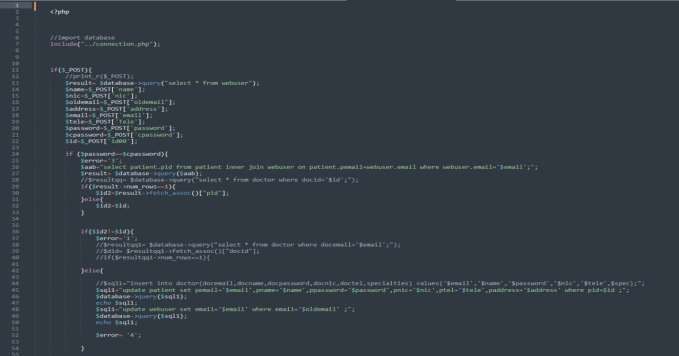
Below we have attached screenshots of php code used in our project.

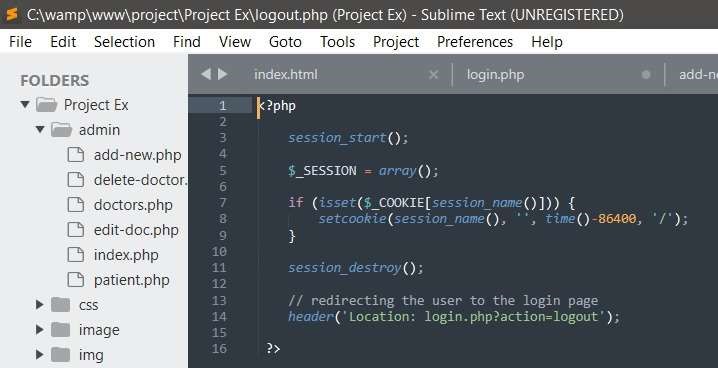






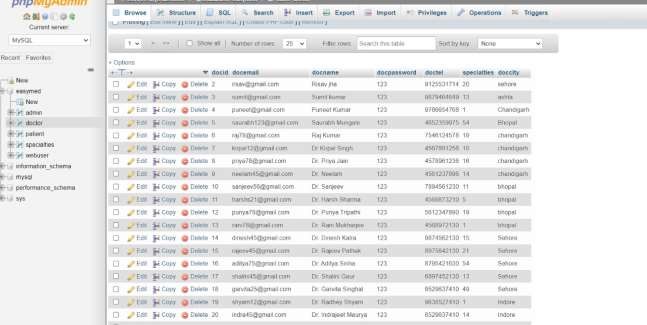


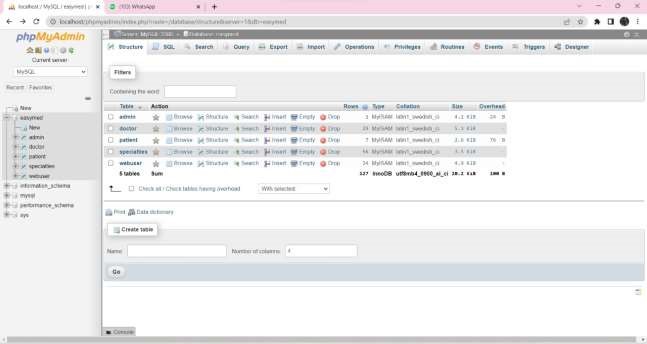


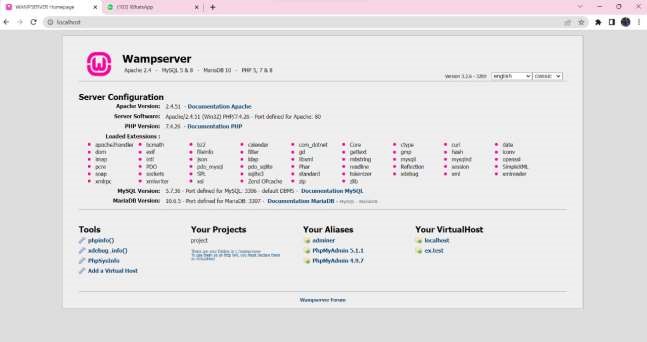


* MySQL-**MySQL** server is a open-source relational database management system which is a major support for web based applications. Databases and related tables are the main component of many websites and applications as the data is stored and exchanged over the web. MySQL server is used for data operations like querying, sorting, filtering, grouping, modifying and joining the tables. Before learning the commonly used queries, let us look into some of the advantage ofMySQL.

The below screenshots shows our database built using MySQL on wamp server.







**CHAPTER 6**

**TESTING AND DEPLOYMENT**

While testing our web application, we have considered the below mentioned template. The below mentioned checklist is almost applicable for all types of web applications depending on their requirements.

**6.1 Usability Testing**

**What is Usability Testing?** Usability testing is nothing but the User-friendliness check.

* In Usability testing, the application flow is tested so that a new user can understand the application easily.
* Basically, system navigation is checked in Usability testing.

**What is the purpose or Goal of Usability testing?**

A Usability test establishes the ease of use and effectiveness of a product using a standard Usability test practice.

**Example Usability Test Cases**

* Web page content should be correct without any spelling or grammatical errors
* All fonts should be same as per the requirements.
* All the text should be properly aligned.
* All the error messages should be correct without any spelling or grammatical errors and the error message should match with the field label.
* Tool tip text should be there for every field.
* All the fields should be properly aligned.
* Enough space should be provided between field labels, columns, rows, and error messages.
* All the buttons should be in a standard format and size.
* Home link should be there on every single page.
* Disabled fields should be grayed out.
* Check for broken links and images.
* Confirmation message should be displayed for any kind of update and delete operation.
* Check the site on different resolutions (640 x 480, 600×800 etc.?)
* Check the end user can run the system without frustration.
* Check the tab should work properly.
* Scroll bar should appear only if required.
* If there is an error message on submit, the information filled by the user should be there.
* Title should display on each web page
* All fields (Textbox, dropdown, radio button, etc) and buttons should be accessible by keyboard shortcuts and the user should be able to perform all operations by using keyboard.
* Check if the dropdown data is not truncated due to the field size. Also, check whether the data is hardcoded or managed via administrator.

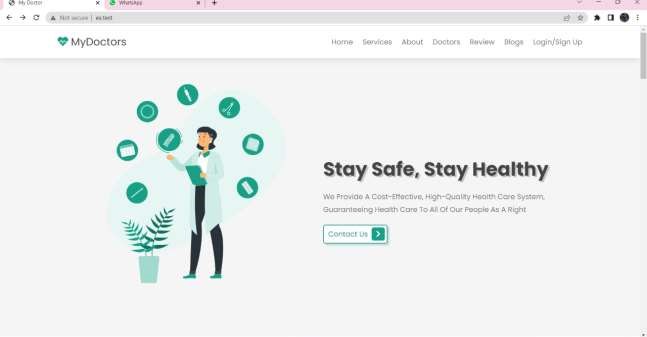


Fig 6.1

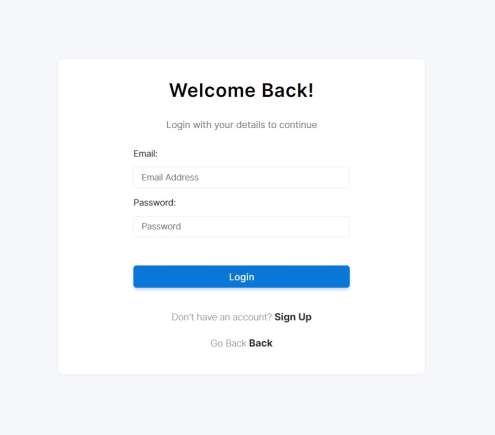


Fig 6.2

**6.2 Functional Testing**

**What is Functional Testing?**

* Testing the features and operational behaviour of a product to ensure they correspond to its specifications.
* Testing that ignores the internal mechanism of a system or component and focuses solely on the outputs generated in response to selected inputs and execution conditions.

**What is the purpose or Goal of Functional testing?**

* The goal of [Functional Testing](https://www.guru99.com/functional-testing.html) is to verify whether your product meets the intended functional specifications mentioned in your development documentation.

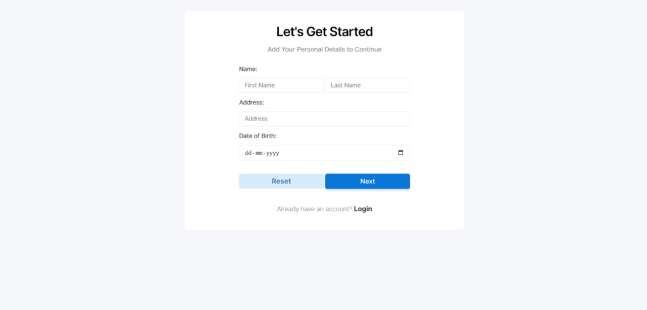
**Example Functional Test Scenarios:**

* Test all the mandatory fields should be validated.
* Test the asterisk sign should display for all the mandatory fields.
* Test the system should not display the error message for optional fields.
* Test that leaps years are validated correctly & do not cause errors/miscalculations.
* Test the numeric fields should not accept the alphabets and proper error message should display.
* Test for negative numbers if allowed for numeric fields.
* Test division by zero should be handled properly for calculations.
* Test the max length of every field to ensure the data is not truncated.
* Test the pop-up message (“This field is limited to 500 characters”) should display if the data reaches the maximum size of the field.
* Test that a confirmation message should display for update and delete operations.
* Test the amount values should display in currency format.
* Test all input fields for special characters.
* Test the timeout functionality.
* Test the Sorting functionality.
* Test the functionality of the buttons available
* Test the Privacy Policy & FAQ is clearly defined and should be available for users.
* Test if any functionality fails the user gets redirected to the custom error page.

Test all the uploaded documents are opened properly.

* Test the user should be able to download the uploaded files.  Test the email functionality of the system.
* Test the [Java s](https://www.guru99.com/java-tutorial.html)cript is properly working in different browsers (IE, Firefox, Chrome, safari and Opera).
* Test to see what happens if a user deletes cookies while in the site.
* Test to see what happens if a user deletes cookies after visiting a site.
* Test all the data inside combo/list box is arranged in chronological order.

Fig 6.3



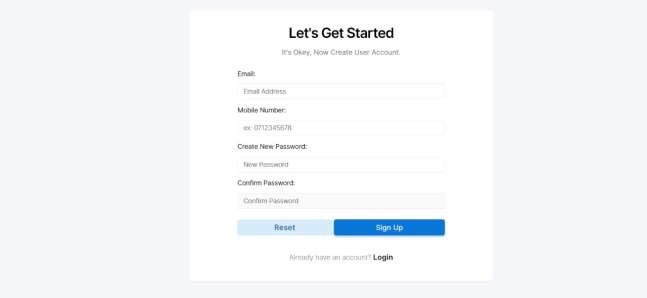


Fig 6.4

**6.3 Compatibility Testing**

**What is Compatibility testing?**

* Compatibility testing is used to determine if your software is compatible with other elements of a system with which it should operate, e.g., Browsers, Operating Systems, or hardware.

**What is the purpose or Goal of Compatibility testing?**

* The purpose of Compatibility testing is to evaluate how well software performs in a particular browser, Operating Systems, hardware or software.

**Sample Compatibility Test Scenarios:**

Test the website in different browsers (IE, Firefox, Chrome, Safari and Opera) and ensure the website is displaying properly.

* Test the HTML version being used is compatible with appropriate browser versions.
* Test the images display correctly in different browsers.
* Test the fonts are usable in different browsers.
* Test the java script code is usable in different browsers.

Fig6.5



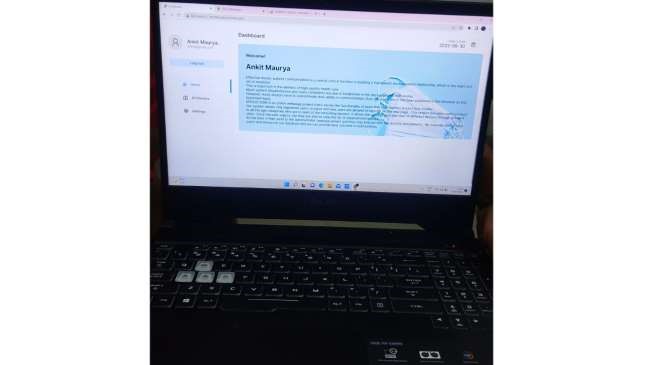


Fig 6.5

**6.4 Database Testing**

**What is Database Testing?**

* In Database testing backend records are tested which have been inserted through the web or desktop applications. The data which is displaying in the web application should match with the data stored in the Database.

**To perform the Database testing, the tester should be aware of the below mentioned points**:

* The tester should understand the functional requirements, business logic, application flow and database design thoroughly.

The tester should figure out the tables, triggers, store procedures, views and cursors used for the application.

* The tester should understand the logic of the triggers, store procedures, views and cursors created.
* The tester should figure out the tables which get affected when insert update and delete (DML) operations are performed through the web or desktop applications. **With the help of the above-mentioned points, the tester can easily write the test scenarios for Database testing.**

**Example Test Cases for Database Testing:**

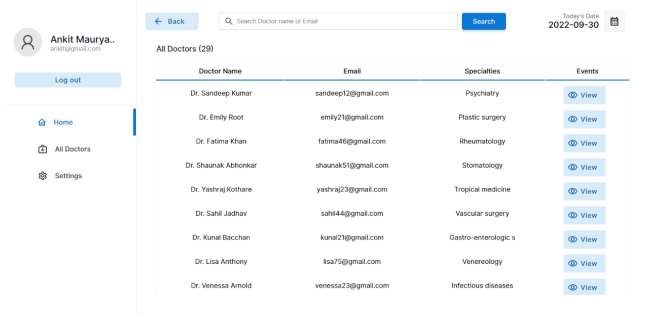
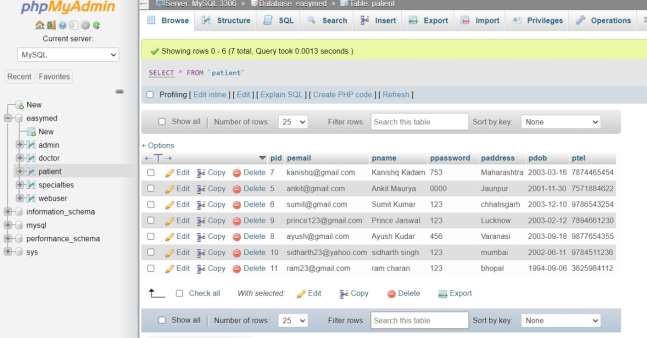
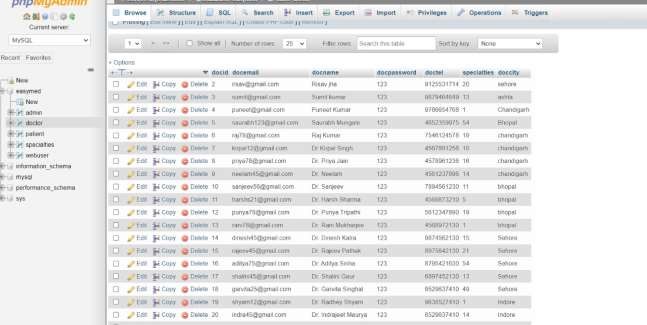
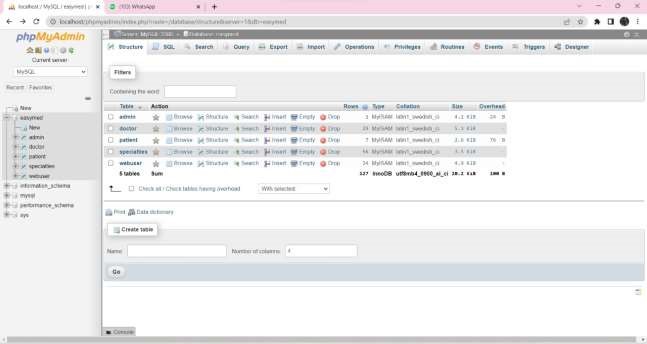
* Verify the database name: The database name should match with the specifications.
* Verify the Tables, columns, column types and defaults: All things should match with the specifications.
* Verify whether the column allows a null or not.
* Verify the Primary and foreign key of each table.
* Verify the Stored Procedure:
* Test whether the Stored procedure is installed or not.
* Verify the Stored procedure name
* Verify the parameter names, types and number of parameters.

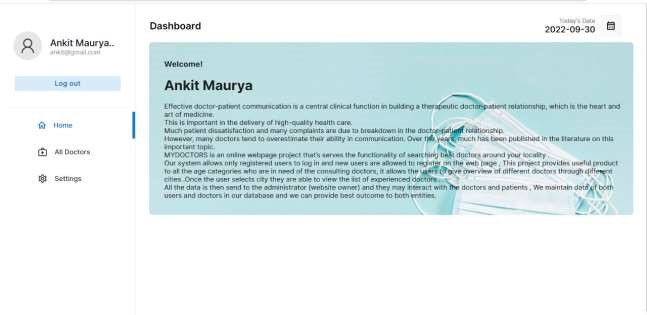
Test the parameters if they are required or not. Test the stored procedure by deleting some parameters

* Test when the output is zero, the zero records should be affected.
* Test the stored procedure by writing simple [SQL q](https://www.guru99.com/sql.html)ueries.
* Test whether the stored procedure returns the values Test the stored procedure with sample input data.
* Verify the behaviour of each flag in the table.
* Verify the data gets properly saved into the database after each page submission.
* Verify the data if the DML (Update, delete and insert) operations are performed.
* Check the length of every field: The field length in the back end and front end must be same.
* Verify the database names of QA, UAT and production. The names should be unique.
* Verify the encrypted data in the database.
* Verify the database size. Also test the response time of each query executed.
* Verify the data displayed on the front end and make sure it is same in the back end.

Verify the data validity by inserting the invalid data in the database.

* Verify the Triggers.





**6.5 Deployment**

The**Deployment Phase of our project**includes the work necessary to deploy the final solution into the target production environments. Plus, creating guides for installation, system operations, system administration, and end-user functionality. Besides, you need to create a detailed plan for implementing a solution across the organization. The production implementation plan is more critical when deploying the solution across several environments maintained by different organizations.

**CHAPTER 7**

### FUTURE ENHANCEMENTS AND CONCLUSION

**6.1**

* **Future Enhancements**

As for future we have vision of expanding and enhancing our existing project. We will achieve this by adding new features such as:

⮚ Consulting doctors online.

⮚ Booking appointment for offline hospital check-up.

⮚ Adding premium add on subscription based services for doctors and hospitals.

⮚ Health related content to get high amount of regular traffic on our website.

**6.2**

* **Conclusion**

⮚ The ultimate objective of any doctor-patient communication is to improve the patient's health and medical care.

⮚ As a result, our website allows visitors to view doctor information and obtain contact information.

⮚ Future plans call for the launch of an online store selling health-related products.

⮚ This website will also be beneficial to people of all ages.

#### **REFERENCES**

1) YouTube

[:https://youtube.com/playlist?list=PLfqMhTWNBTe3H6c9OGXb5\_6wcc1Mca52n](https://youtube.com/playlist?list=PLfqMhTWNBTe3H6c9OGXb5_6wcc1Mca52n)

2)MDN Web:

<https://developer.mozilla.org/en-US/docs/Web/HTML><https://developer.mozilla.org/en-US/docs/Web/CSS><https://developer.mozilla.org/en-US/docs/Web/JavaScript>

3)Udemy

<https://www.udemy.com/course/the-complete-web-development-bootcamp/>