Table of Contents

Mobile E Health Care – an Web application for medical data maintenance

[“E Health Care ” – An Web Application for Medical Data Maintenance 3](#_Toc124343007)

[ABSTRACT 3](#_Toc124343008)

[CHAPTER 1 – INTRODUCTION 4](#_Toc124343009)

[1.1 System Specifications 5](#_Toc124343010)

[CHAPTER 2 – LITERATURE REVIEW 6](#_Toc124343011)

[CHAPTER 3 OVERALL DESCRIPTION OF THE PROPOSED SYSTEM 10](#_Toc124343012)

[3.1 Module List 10](#_Toc124343013)

[3.2Module Description 11](#_Toc124343014)

[CHAPTER 4 – DESIGN 14](#_Toc124343015)

[4.1UML Diagrams: 14](#_Toc124343016)

[**4.1.1Usecase Diagrams**: 15](#_Toc124343017)

[**4.1.2 Sequence Diagram:** 17](#_Toc124343018)

[**4.1.3 Collaboration Diagram:** 19](#_Toc124343019)

[**4.1.4 Class Diagram:** 20](#_Toc124343020)

[**4.1.4 Architecture Design** 21](#_Toc124343021)

[**4.1.5. Data Flow Diagram** 23](#_Toc124343022)

[**4.1.6Table Design** 25](#_Toc124343023)

[**4.1.8 ER DIAGRAM** 25](#_Toc124343024)

[**4.1.9 Workflow Diagram** 26](#_Toc124343025)

[**4.1.10 Activity Diagram** 27](#_Toc124343026)

[CHAPTER 5 - OUTPUT SCREENSHOTS 29](#_Toc124343027)

[CHAPTER 6 – IMPLEMENTATION DETAILS 30](#_Toc124343028)

[6.1 Introduction to Html Framework 30](#_Toc124343029)

[6.2 Cascading Style Sheets (CSS) 33](#_Toc124343030)

[6.3 MYSQL Server 38](#_Toc124343031)

[6.4PHP 39](#_Toc124343032)

[6.5 ANGULAR JAVA SCRIPT 41](#_Toc124343033)

[CHAPTER 8-TECHNICAL FEASIBILITY 45](#_Toc124343034)

[CHAPTER 9-SYSTEM TESTING 47](#_Toc124343035)

[CHAPTER 10– CONCLUSIONS 55](#_Toc124343036)

[CHAPTER 11- REFERENCES 56](#_Toc124343037)

[CHAPTER 12 – Coding 58](#_Toc124343038)

# “E Health Care ” – An Web Application for Medical Data Maintenance

# ABSTRACT

The advent of technology in our everyday lives has fundamentally altered the way by which the information is collected, interpreted, stored, and analyzed. Data networks in the developing world have been an important part of health-care systems. Moreover, due to a growth in population, the journey of the patient across the multiple routes in the healthcare system would highly challenge the accountability, tracking, recording, and analysis of patient data. Developing a model for gathering patient medical information from various health care contexts (i.e., family physicians, private clinics, emergency department visits, routes, and hospital systems) is essential, and requires more space to store all patient data. The cloud database is an efficient solution to implement this application and it consists of various services like authentication, storage. The users are authenticated and then the user data is maintained in a centralized manner via mysql. This research work also incorporates a Quick Response (QR code) to access patient id. The qr code will serve as a sharing media with other health care providers and facilitated decision making. It helps to make good decisions before treatment by checking previous medical records of the patient.

Keywords: Medical Records, Cloud, QR Code, Smart Card.

# CHAPTER 1 – INTRODUCTION

The automation systems are implemented in hospitals and medical centers to provide an optimal working atmosphere for healthcare professionals. One of the main roles of such systems is quick access to accurate health data. Such programs should provide a network of information that attains processes and stores information for the patient to fulfill those specifications. There can be several sources from which patient-related information can be obtained: the patient himself, patient -related test reports, electronic patient tracking services, patient-related assessments by doctors and patient records already stored. Besides the symptoms of suspected disease, the physician needs to be able to retrieve the patient's previous specifics during his examination. The standard way to get specific medical data is by linking to the hospital portal. Due to the high data rate, simultaneous access to databases from various terminals located in the examination rooms can cause performance problems in some circumstances. For other situations, patient data may be required in an area where there is no network communication facility (for example, an ambulance) or at another hospital where there is no registered patient. The use of smart storage and recovery techniques to increase the capabilities of hospital automation systems will solve these problems. Transportable media may play a key role in communicating a small number of patients-which can provide useful data for a hospital automation program in practice. The media selected for the above-mentioned purposes should be affordable, simple to use, interact, secure and update with new information and it should not damage quickly. Once these criteria are considered, "smart card" tends to be the most appropriate tool to be found in information systems for healthcare. Medical information is an ever-increasing source of information provided by hospitals consisting of patient records in the process of paper work that can also be processed simpler and faster with Quick Response Code (QRC) technology. Our main goal is to develop a Smart Health Card Web application that will provide services such as Medical Reports, Prescriptions, Medical History of the user, and Smart card with generated QR code for each user based on their updated information on the cloud database. It is very error-prone for hospital ad ministrations to keep track o f a ll the health activities and records. It's a method that is very costly, and time-consuming. The collection and maintenance of various medical data are incredibly unreliable. Even maintaining the records on paper is not so economically and technically feasible. Through this system, low cost and efficient automat ion of the current system will be ensured because in our daily life every smart phone user is ma king payments digitally by using QR code technology. Hence, people are very much interactive with QR code-based applications. Likewise, this application is also implemented with QR code technology to share and access medical data and reports of any user by securely scanning QR code. User health information and reports are automatically shared and accessed by any doctor or healthcare provider using the QR code image. Therefore, the doctor then scans the patient's unique QR code image with his/her Web mobile and it’s focused on the health data of the patient, prescribes medicines, and related advice that is notified to the user application. Whenever medicines are required for m pharmacy shops, the pharmacist able to scan the user QR code and gives medicines accordingly.

## System Specifications

**Hardware Requirements:-**

Processor : Intel 3

Installed memory (RAM) : 4 GB

Hard Disk : 500 GB

Operating System : Windows 7,8,10 - 64 bit

**Software Requirements: -**

Software Requirements: -

Front End: HTML5, CSS3, Bootstrap

Back End: PHP 7.4, MYSQL

Control End: Angular Java Script

Tools:

xampp-win64-7.4

# 

# CHAPTER 2 – LITERATURE REVIEW

Jacek Ruminski et al. [1] has projected an application that is implemented by using the QR code to exchange laboratory results. It is proposed that secure data should be exchanged between the patient and the laboratory and also between EHR (E-Health Records) and the patient, but there is a need for interaction between healthcare professionals and patients, as healthcare professionals can access and update on patient information in response. Paschou Mersini et al. [2] has implemented an interactive system designed to be used by healthcare workers in healthcare facilities implemented on standard smart phones, laptops, and mobile devices. The goal here is to allow the use of standard smart phones by health care providers, nursing staff, and professionals involved in the facility , irrespective of the existence of wireless networks in the environment. P. Geethika Choudary et al. [3] has implemented an application which consists of health trackers, convertors, mental health awareness platform, and local maps. The users are allowed to interact with those modules and process the health assessments. Sudha.G et al. [4] developed a model, for patient multimedia medical data monitoring while the user is on the move. Due to the lack o f less storage in mobile devices to store data they used My SQL database as a server system for storing and accessing large data with security. It provides authentication and security and accessing the database. Mungyu Bae et al. [5] implemented a strategy for reducing the time -consuming process of writing papers for medical visits to medical centers, so a local hospital can handle patient personal information cost-effectively. Through this approach, the patient has to mention his/her symptoms to the doctor each time. Mr. Mahale Kishor M et al. [6] exhibited a method using, Naive Bays and Decision tree algorithms those are implemented with and without using PCA on the prediction dataset for heart disease. Chayakrit Charoensiriwath et al. [7] exhibited a low-cost model that was designed and tested in a limited budget for the public hospital. The whole system used QR code and mobile devices to track as -is hospital operations and time spent on patient activities. With all the available activity data, two algorithms are built to identify two quantities that are useful for quality improvement: the most time -consuming and bottleneck activities. Rizal Mohd Nor et al. [8 ] developed a mobile pharmaceutical adherence network it will allow the government and community hospitals to resolve some of the issues related to compliance with medicines in Malaysia. The proposed solution is built with the Web platform which uses the camera (scanner element) to scan the QR code which is labeled on medicines. Jafar Saniie et al. [9 ] implemented a Mobile Health Monitoring system for patients , which utilizes multiple biosensors are anticipated patients’ signs so that, in addition to the biosensors, the device can easily obtain medicine using the form of a sensor known as the environmental sensor to enhance accuracy and analysis. Daryl Abel et al. [10] presented a method for repairing a doctor’s or medical consultant’s appointment. Using this method, patients can easily use a Smartphone application to fix appointments. A mobile app is being created and a road map for further analysis is also included.

**2.1Existing Solution:**

In an existing system for recording health or medical data is write down in a paper and summarize data in a book. Filing, archiving and also tracing the records is the big issue. Time consumption is needed where looking for a medical record paper if there is big number of records. On the other side, it is also difficult to bring the medical record paper if the sick man refers to others hospital or doctor for more medical treatment. Due to those issue its need more feasible medium for health and medical record.

It is difficult to collect the history of data in the integrated form. It is especially important when a patient is mobile, using health services from different providers. Health and medical data especially about records of medical check is very crucial data. These data are used as reference for medical treatment. Wrong treatment because wrong reference, it is something should never occur. Security method is needed in order to avoid non authorize people can change the data.

**2.2 Proposed Solution:**

The proposed system has integrated with various services (medical reports, prescriptions, medical bills, and medical history) those are easy to use and will contribute as a useful model for personal health monitoring. To this end, this research work creates an Web application with cloud storage that includes the login verification to authenticate each user to access their account, containing all personal data and health records. The data will then be stored in the cloud database, and a QR-coded smart card will be created in the profile containing the proper information.

The smart card may be scanned in case of emergencies, and the information contained in the cloud will be retrieved It saves time for a patient admitted to a hospital to begin treatment. It saves time to complete the medical procedures and get the patient and start operating. This is also safe and stable for storing and retrieving data. When using this approach, it not only saves the victim's life but also allows the doctors to operate safely and it also reduces medical errors. After the treatment of the patient, the doctor/healthcare provider can update the condition and send precautions to the patient application as a notification.

# CHAPTER 3 OVERALL DESCRIPTION OF THE PROPOSED SYSTEM

## 3.1 Module List

**Patient**

* Register
* Login
* List of Doctor
* Book Doctor
* View Medical Reports and Bills
* List of Medical Reports
* Medical Prescriptions
* Medical Information
* Medical History
* Smart Health Card
* My profile

**Doctor**

* Register
* Login
* Generate QR Code (Web browser)
* Scan QR Code (QR Scanner – 3rd party)
* Create hospitals
* Update Details
* Booking Details
* Update Medical Reports and Bills
* Update Medical Prescriptions
* Update Medical Information
* Update Medical History
* View Health Card

## 3.2Module Description

The system after careful maintain the records has been identified to be presented with the following modules **Patient and Doctor.**

**Patient**

* **Register**

Patient has to register their basic details to get access with this application service.

* **Login**

Once they have registered they need to login to avail the service at the needy time.

* **List of Doctor**

Logging in with the application will provide you the lists of doctors checked by the patient.

* **Book Doctor**

So that the patient can search and select the doctors among the list and booking the doctor according to their place and time.

* **View Medical Reports and Bills**

The patient store and access the medical records which they have stored in the cloud database and also share those records with other users whenever they required.

* **Medical Prescriptions**

The user has been allowed to store various medical prescript ions that are given by doctors/health care providers. So me people don’t remember all the prescriptions in their busy schedule, in that case, the user can be store and access the prescriptions from anywhere and purchase medicines by sharing prescriptions to clinical or pharmacists.

* **Medical Information**

The personal information of a user by manual entry including basic health condition details. There are various allergies and diseases which are frequent to some people, those details are to be considered as medical history, this data very helpful for healthcare providers while emergency. Medical history data will be very useful to identify the user when he/she is in any medical necessity.

* **Smart Health Card**

Using smart card as data records medium, the first thing to do is define what kind of data to be stored in order to count the smart card capacity needed. Thus, find out the smart card which accomplish with size of capacity as defined before. The smart card on the market might be not fit in with the capacity defined. So find out the smart card which bigger than capacity needed or redefined the data to be stored. It is also should be considered that smart card with bigger size of capacity is costly.

* **My profile**

The patient can view and modify the their profile and maintain the their profile details

**Doctor**

* **Register**

At first every Doctor has to register their details with the admin for getting approval.

* **Login**

Registered Doctors can login their accounts if they got their approval from the admin.

* **Generate QR Code**

The consulate doctor has to generate the QR code for the patients. The use of QR code easily can access the particular patients details

* **Scan QR Code**

After creating the QR code, Doctors can scan the QR code they access the details from particular patients.

* **Create hospitals**

The doctors can create an their working hospitals details.

* **Update details**

The doctor will update the patient medicine recommended details.

* **Booking Details**

The doctor will booking their patient details.

* **Update Medical Reports and Bills**

The doctor will update the patient medical reports and their bills also.

* **Update Medical Prescriptions**

The doctor will update the patient medical prescriptions and medical reports also.

* **Update Medical Information**

The doctor will update the patient details, patient history, patient’s health conditions and medical updated prescription are updated in medical information.

* **Update Medical History**

The doctor will update and maintain the patient history collectively patient’s current stage of history also updated in patient medical history.

* **View Health Card**

The doctor can view the patient’s health card.

**3.2 System Features**

In the life of the software development, problem analysis provides a base for design and development phase. The problem is analyzed so that sufficient matter is provided to design a new system. Large problems are sub-divided into smaller once to make them understandable and easy for finding solutions. Same in this project all the task are sub-divided and categorized.

# CHAPTER 4 – DESIGN

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software.

The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer’s requirements into finished software or a system.

Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data.

## 4.1UML Diagrams:

UML stands for Unified Modeling Language. UML is a language for specifying, visualizing and documenting the system. This is the step while developing any product after analysis. The goal from this is to produce a model of the entities involved in the project which later need to be built. The representation of the entities that are to be used in the product being developed need to be designed.

There are various kinds of methods in software design:

* Use case Diagram
* Sequence Diagram
* Collaboration Diagram
* Class Diagram

**4.1.1Usecase Diagrams**:

Use case diagrams model behavior within a system and helps the developers understand of what the user require. The stick man represents what’s called an actor. Use case diagram can be useful for getting an overall view of the system and clarifying that can do and more importantly what they can’t do.





Use case diagram consists of use cases and actors and shows the interaction between the use case and actors.

* The purpose is to show the interactions between the use case and actor.
* To represent the system requirements from user’s perspective.
* An actor could be the end-user of the system or an external system.

**4.1.2 Sequence Diagram:**

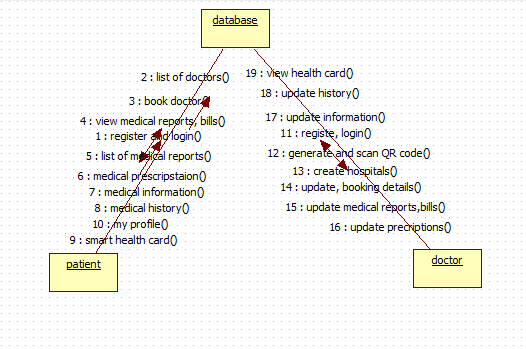
Sequence diagram and collaboration diagram are called INTERACTION DIAGRAMS. An interaction diagram shows an interaction, consisting of set of objects and their relationship including the messages that may be dispatched among them.

A sequence diagram is an introduction that empathizes the time ordering of messages. Graphically a sequence diagram is a table that shows objects arranged along the X-axis and messages ordered in increasing time along the Y-axis.



**4.1.3 Collaboration Diagram:**

A **collaboration diagram** is a type of visual presentation that shows how various software objects interact with each other within an overall IT architecture and how users can benefit from this **collaboration**. A **collaboration diagram** often comes in the form of a visual chart that resembles a flow chart.



**4.1.4 Class Diagram:**



**4.1.4 Architecture Design**

Patient

List and book doctor

List medical reports , prescriptions

Medical information, hsitory

Smart health card

My profile

View medical reports, bills

Database

Doctor

Generate, scan QR code

Create hospitals

Update, booking details

Update medical reports, prescription

Update information, history

database

View health card

**4.1.5. Data Flow Diagram**

**Patient**

List and book doctor

View medical reports, bills

My profile

Smart health card

Medical presccriptions, information, history

List medical reports

**Doctor**

Generate, scan QR code

Create hospitals

Update, booking details

Update medical reports, bill

Medical prescription, information, history

View health card

**4.1.6Table Design**

**User Register & Login**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| User ID | Name | Email Id | Password | Mobile | Address | City | Question 1 | Question 2 |
|  |  |  |  |  |  |  |  |  |

**Doctor Register & Login**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| User ID | Name | Email Id | Password | Mobile | Address | City | Question 1 | Question 2 |
|  |  |  |  |  |  |  |  |  |

**HosptialDetails**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| User ID | Hosptial Name | Treatment | Doctor | Time | Locality | Address | City | Services |
|  |  |  |  |  |  |  |  |  |

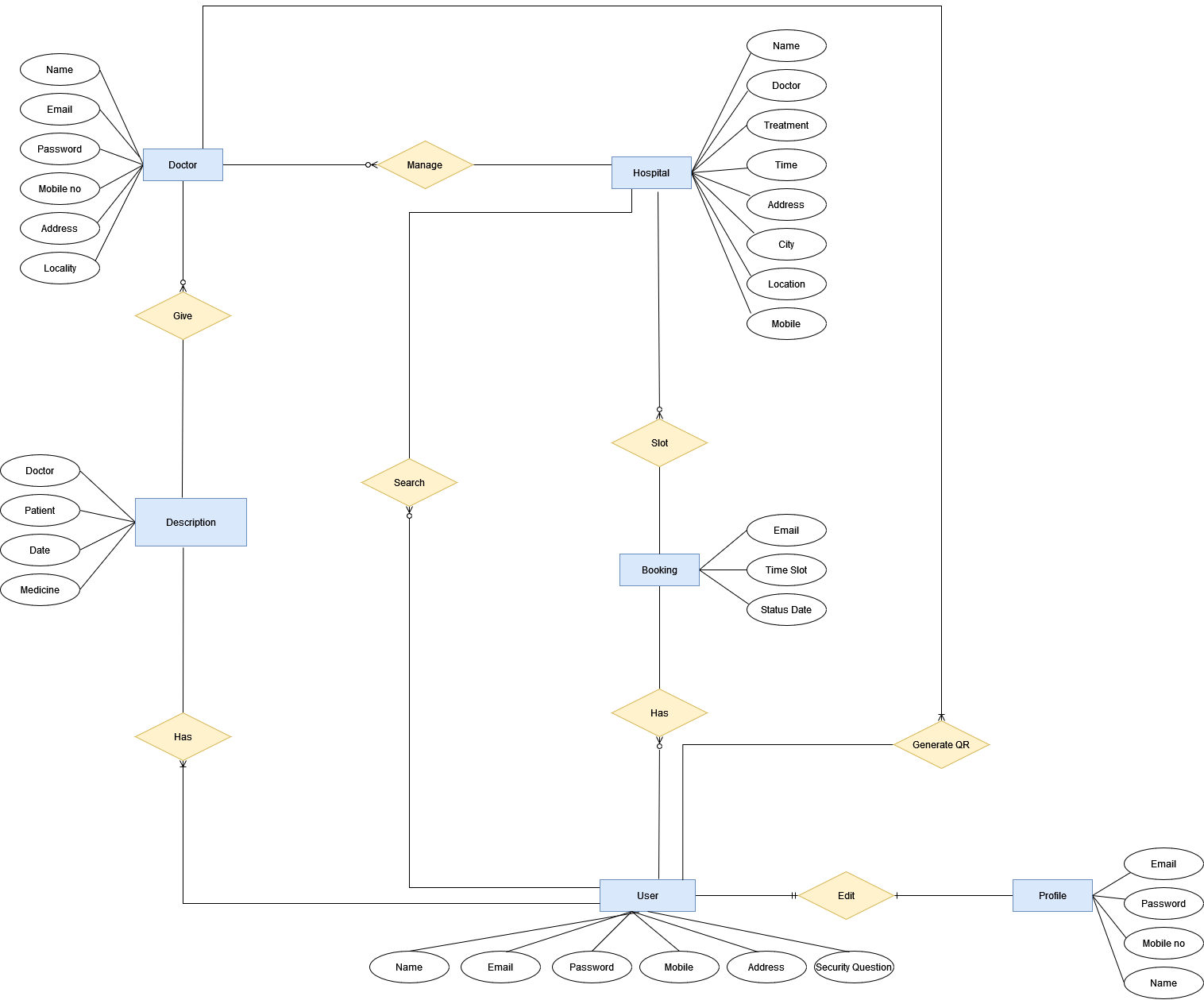
**HosptialBooking**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| User ID | Hosptial Name | Treatment | Doctor | Time |
|  |  |  |  |  |

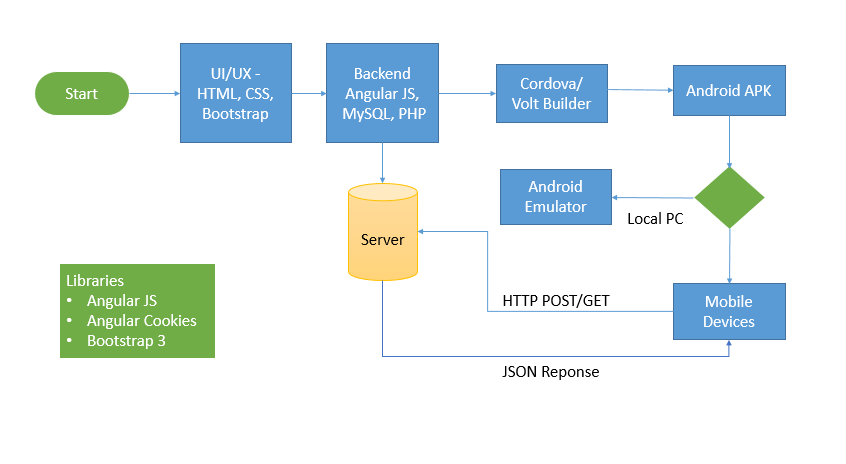
**Medicine Prescription**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| User ID | Date | Medicine 1 | Medicine 2 | Medicine 3 | Notes |
|  |  |  |  |  |  |

**4.1.8 ER** **DIAGRAM**



**4.1.9 Workflow Diagram**



**4.1.10 Activity Diagram**



# CHAPTER 5 - OUTPUT SCREENSHOTS

# 

# CHAPTER 6 – IMPLEMENTATION DETAILS

## 6.1 Introduction to Html Framework

Hyper Text Markup Language, commonly referred to as HTML, is the standard [markup language](https://en.wikipedia.org/wiki/Markup_language) used to create [web pages](https://en.wikipedia.org/wiki/Web_page). Along with [CSS](https://en.wikipedia.org/wiki/Cascading_Style_Sheets), and [JavaScript](https://en.wikipedia.org/wiki/JavaScript), HTML is a cornerstone technology used to create web pages, as well as to create user interfaces for mobile and [web applications](https://en.wikipedia.org/wiki/Web_applications). [Web browsers](https://en.wikipedia.org/wiki/Web_browser) can read HTML files and render them into visible or audible web pages. HTML describes the structure of a [website](https://en.wikipedia.org/wiki/Website) [semantically](https://en.wikipedia.org/wiki/Semantic) along with cues for presentation, making it a markup language, rather than a [programming language](https://en.wikipedia.org/wiki/Programming_language).

[HTML elements](https://en.wikipedia.org/wiki/HTML_element) form the building blocks of HTML pages. HTML allows [images](https://en.wikipedia.org/wiki/Img_(HTML_element)) and other objects to be embedded and it can be used to create [interactive forms](https://en.wikipedia.org/wiki/Fieldset). It provides a means to create [structured documents](https://en.wikipedia.org/wiki/Structured_document) by denoting structural semantics for text such as headings, paragraphs, lists, [links](https://en.wikipedia.org/wiki/Hyperlink), quotes and other items. HTML elements are delineated by tags, written using [angle brackets](https://en.wikipedia.org/wiki/Bracket#Angle_brackets). Tags such as <img /> and <input /> introduce content into the page directly. Others such as <p>...</p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed [scripts](https://en.wikipedia.org/wiki/Scripting_language) written in languages such as [JavaScript](https://en.wikipedia.org/wiki/JavaScript) which affect the behavior of HTML web pages. HTML markup can also refer the browser to [Cascading Style Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets) (CSS) to define the look and layout of text and other material.

HyperText Markup Language (HTML) is the standard [markup language](https://en.wikipedia.org/wiki/Markup_language) for creating [web pages](https://en.wikipedia.org/wiki/Web_page) and [web applications](https://en.wikipedia.org/wiki/Web_application). With [Cascading Style Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets) (CSS) and [JavaScript](https://en.wikipedia.org/wiki/JavaScript) it forms a triad of cornerstone technologies for the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web).[[1]](https://en.wikipedia.org/wiki/HTML#cite_note-1)[Web browsers](https://en.wikipedia.org/wiki/Web_browser) receive HTML documents from a [webserver](https://en.wikipedia.org/wiki/Webserver) or from local storage and render them into multimedia web pages. HTML describes the structure of a web page [semantically](https://en.wikipedia.org/wiki/Semantic) 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/Img_%28HTML_element%29) and other objects, such as [interactive forms,](https://en.wikipedia.org/wiki/Fieldset) may be embedded into the rendered page. It provides a means to create [structured documents](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. HTML elements are delineated by tags, written using [angle brackets](https://en.wikipedia.org/wiki/Bracket#Angle_brackets). Tags such as <img /> and <input /> introduce content into the page directly. Others such as <p>...</p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a [scripting language](https://en.wikipedia.org/wiki/Scripting_language) such as [JavaScript](https://en.wikipedia.org/wiki/JavaScript) which affect the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The [World Wide Web Consortium](https://en.wikipedia.org/wiki/World_Wide_Web_Consortium) (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.[[2]](https://en.wikipedia.org/wiki/HTML#cite_note-deprecated-2)

In 1980, physicist [Tim Berners-Lee](https://en.wikipedia.org/wiki/Tim_Berners-Lee), a contractor at [CERN](https://en.wikipedia.org/wiki/CERN), proposed and prototyped [ENQUIRE](https://en.wikipedia.org/wiki/ENQUIRE), a system for CERN researchers to use and share documents. In 1989, Berners-Lee wrote a memo proposing an [Internet](https://en.wikipedia.org/wiki/Internet)-based [hypertext](https://en.wikipedia.org/wiki/Hypertext) system.[[3]](https://en.wikipedia.org/wiki/HTML#cite_note-3) Berners-Lee specified HTML and wrote the browser and server software in late 1990. That year, Berners-Lee and CERN data systems engineer [Robert Cailliau](https://en.wikipedia.org/wiki/Robert_Cailliau) collaborated on a joint request for funding, but the project was not formally adopted by CERN. In his personal notes[[4]](https://en.wikipedia.org/wiki/HTML#cite_note-4) from 1990 he listed[[5]](https://en.wikipedia.org/wiki/HTML#cite_note-5) "some of the many areas in which hypertext is used" and put an encyclopedia first.

The first publicly available description of HTML was a document called "HTML Tags", first mentioned on the Internet by Tim Berners-Lee in late 1991.[[6]](https://en.wikipedia.org/wiki/HTML#cite_note-tagshtml-6)[[7]](https://en.wikipedia.org/wiki/HTML#cite_note-7) It describes 18 elements comprising the initial, relatively simple design of HTML. Except for the hyperlink tag, these were strongly influenced by [SGMLguid](https://en.wikipedia.org/wiki/SGMLguid), an in-house [Standard Generalized Markup Language](https://en.wikipedia.org/wiki/Standard_Generalized_Markup_Language) (SGML)-based documentation format at CERN. Eleven of these elements still exist in HTML 4.[[8]](https://en.wikipedia.org/wiki/HTML#cite_note-8)

HTML is a [markup language](https://en.wikipedia.org/wiki/Markup_language) that [web browsers](https://en.wikipedia.org/wiki/Web_browser) use to interpret and [compose](https://en.wikipedia.org/wiki/Typesetting) text, images, and other material into visual or audible web pages. Default characteristics for every item of HTML markup are defined in the browser, and these characteristics can be altered or enhanced by the web page designer's additional use of [CSS](https://en.wikipedia.org/wiki/Cascading_Style_Sheets). Many of the text elements are found in the 1988 ISO technical report TR 9537 Techniques for using SGML, which in turn covers the features of early text formatting languages such as that used by the [RUNOFF command](https://en.wikipedia.org/wiki/TYPSET_and_RUNOFF) developed in the early 1960s for the [CTSS](https://en.wikipedia.org/wiki/Compatible_Time-Sharing_System) (Compatible Time-Sharing System) operating system: these formatting commands were derived from the commands used by typesetters to manually format documents. However, the SGML concept of generalized markup is based on elements (nested annotated ranges with attributes) rather than merely print effects, with also the separation of structure and markup; HTML has been progressively moved in this direction with CSS.

Berners-Lee considered HTML to be an application of SGML. It was formally defined as such by the [Internet Engineering Task Force](https://en.wikipedia.org/wiki/Internet_Engineering_Task_Force) (IETF) with the mid-1993 publication of the first proposal for an HTML specification, the "Hypertext Markup Language (HTML)" Internet Draft by Berners-Lee and [Dan Connolly](https://en.wikipedia.org/wiki/Dan_Connolly_%28computer_scientist%29), which included an SGML [Document Type Definition](https://en.wikipedia.org/wiki/Document_Type_Definition) to define the grammar.[[9]](https://en.wikipedia.org/wiki/HTML#cite_note-9)[[10]](https://en.wikipedia.org/wiki/HTML#cite_note-10) The draft expired after six months, but was notable for its acknowledgment of the [NCSA Mosaic](https://en.wikipedia.org/wiki/Mosaic_%28web_browser%29) browser's custom tag for embedding in-line images, reflecting the IETF's philosophy of basing standards on successful prototypes.[[11]](https://en.wikipedia.org/wiki/HTML#cite_note-raymond-11) Similarly, [Dave Raggett](https://en.wikipedia.org/wiki/Dave_Raggett)'s competing Internet-Draft, "HTML+ (Hypertext Markup Format)", from late 1993, suggested standardizing already-implemented features like tables and fill-out forms.[[12]](https://en.wikipedia.org/wiki/HTML#cite_note-html.2B-12)

After the HTML and HTML+ drafts expired in early 1994, the IETF created an HTML Working Group, which in 1995 completed "HTML 2.0", the first HTML specification intended to be treated as a standard against which future implementations should be based.[[13]](https://en.wikipedia.org/wiki/HTML#cite_note-13)

Further development under the auspices of the IETF was stalled by competing interests. Since 1996, the HTML specifications have been maintained, with input from commercial software vendors, by the [World Wide Web Consortium](https://en.wikipedia.org/wiki/World_Wide_Web_Consortium) (W3C).[[14]](https://en.wikipedia.org/wiki/HTML#cite_note-raggett-14) However, in 2000, HTML also became an international standard ([ISO](https://en.wikipedia.org/wiki/International_Organization_for_Standardization)/[IEC](https://en.wikipedia.org/wiki/International_Electrotechnical_Commission) 15445:2000). HTML 4.01 was published in late 1999, with further errata published through 2001. In 2004, development began on HTML5 in the [Web Hypertext Application Technology Working Group](https://en.wikipedia.org/wiki/Web_Hypertext_Application_Technology_Working_Group) (WHATWG), which became a joint deliverable with the W3C in 2008, and completed and standardized on 28 October 2014.[[15]](https://en.wikipedia.org/wiki/HTML#cite_note-15)

## 6.2 Cascading Style Sheets (CSS)

CSS is a [style sheet language](https://en.wikipedia.org/wiki/Style_sheet_language) used for describing the [presentation](https://en.wikipedia.org/wiki/Presentation_semantics) of a document written in a [markup language](https://en.wikipedia.org/wiki/Markup_language). Although most often used to set the visual style of [web pages](https://en.wikipedia.org/wiki/Web_page) and user interfaces written in [HTML](https://en.wikipedia.org/wiki/HTML) and [XHTML](https://en.wikipedia.org/wiki/XHTML), the language can be applied to any [XML](https://en.wikipedia.org/wiki/XML) document, including [plain XML](https://en.wikipedia.org/wiki/Plain_Old_XML), [SVG](https://en.wikipedia.org/wiki/Scalable_Vector_Graphics) and[XUL](https://en.wikipedia.org/wiki/XUL), and is applicable to rendering in [speech](https://en.wikipedia.org/wiki/Speech_synthesis), or on other media. Along with HTML and [JavaScript](https://en.wikipedia.org/wiki/JavaScript), CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for [web applications](https://en.wikipedia.org/wiki/Web_applications), and user interfaces for many mobile applications.

CSS is designed primarily to enable [the separation of document content from document presentation](https://en.wikipedia.org/wiki/Separation_of_presentation_and_content), including aspects such as the [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 HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content, such as [semantically insignificant tables](https://en.wikipedia.org/wiki/Tableless_web_design) that were widely used to format pages before consistent CSS rendering was available in all major browsers. CSS makes it possible to separate presentation instructions from the HTML content in a separate file or style section of the HTML file. For each matching [HTML element](https://en.wikipedia.org/wiki/HTML_element), it provides a list of formatting instructions. For example, a CSS rule might specify that "all heading 1 elements should be [bold](https://en.wikipedia.org/wiki/Bold)", leaving pure semantic HTML markup that asserts "this text is a level 1 heading" without formatting code such as a<bold> tag indicating how such text should be displayed.

This separation of formatting and content makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or[screen reader](https://en.wikipedia.org/wiki/Screen_reader)) and on [Braille-based](https://en.wikipedia.org/wiki/Braille_display), tactile devices. It can also be used to display the web page differently depending on the screen size or device on which it is being viewed. Although the author of a web page typically links to a CSS file within the markup file, readers can specify a different style sheet, such as a CSS file stored on their own computer, to override the one the author has specified. If the author or the reader did not link the document to a style sheet, the default style of the browser will be applied. Another advantage of CSS is that aesthetic changes to the [graphic design](https://en.wikipedia.org/wiki/Graphic_design) of a document (or hundreds of documents) can be applied quickly and easily, by editing a few lines in one file, rather than by a laborious (and thus expensive) process of crawling over every document line by line, changing markup.

The CSS specification describes a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities (or weights) are calculated and assigned to rules, so that the results are predictable.

Cascading Style Sheets (CSS) is a [style sheet language](https://en.wikipedia.org/wiki/Style_sheet_language) used for describing the [presentation](https://en.wikipedia.org/wiki/Presentation_semantics) of a document written in a [markup language](https://en.wikipedia.org/wiki/Markup_language).[[1]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-1) Although most often used to set the visual style of [web pages](https://en.wikipedia.org/wiki/Web_page) and user interfaces written in [HTML](https://en.wikipedia.org/wiki/HTML) and [XHTML](https://en.wikipedia.org/wiki/XHTML), the language can be applied to any [XML](https://en.wikipedia.org/wiki/XML) document, including [plain XML](https://en.wikipedia.org/wiki/Plain_Old_XML), [SVG](https://en.wikipedia.org/wiki/Scalable_Vector_Graphics) and [XUL](https://en.wikipedia.org/wiki/XUL), and is applicable to rendering in [speech](https://en.wikipedia.org/wiki/Speech_synthesis), or on other media. Along with HTML and [JavaScript](https://en.wikipedia.org/wiki/JavaScript), CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for [web applications](https://en.wikipedia.org/wiki/Web_applications), and user interfaces for many mobile applications.[[2]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-2)

CSS is designed primarily to enable [the separation of document content from document presentation](https://en.wikipedia.org/wiki/Separation_of_presentation_and_content), including aspects such as the [layout](https://en.wikipedia.org/wiki/Page_layout), [colors](https://en.wikipedia.org/wiki/Color), and [fonts](https://en.wikipedia.org/wiki/Typeface).[[3]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-3) 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 HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Separation of formatting and content makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or [screen reader](https://en.wikipedia.org/wiki/Screen_reader)), and on [Braille-based](https://en.wikipedia.org/wiki/Braille_display) tactile devices. It can also display the web page differently depending on the screen size or viewing device. Readers can also specify a different style sheet, such as a CSS file stored on their own computer, to override the one the author specified.

Changes to the [graphic design](https://en.wikipedia.org/wiki/Graphic_design) of a document (or hundreds of documents) can be applied quickly and easily, by editing a few lines in the CSS file they use, rather than by changing markup in the documents.

The CSS specification describes a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities (or weights) are calculated and assigned to rules, so that the results are predictable.

The CSS specifications are maintained by the [World Wide Web Consortium](https://en.wikipedia.org/wiki/World_Wide_Web_Consortium) (W3C). Internet media type ([MIME type](https://en.wikipedia.org/wiki/MIME_media_type)) text/css is registered for use with CSS by [RFC 2318](https://tools.ietf.org/html/rfc2318) (March 1998). The W3C operates a free [CSS validation service](https://en.wikipedia.org/wiki/W3C_Markup_Validation_Service#CSS_validation) for CSS documents.

In CSS, selectors declare which part of the markup a style applies to by matching tags and attributes in the markup itself.

Selectors may apply to:

all [elements](https://en.wikipedia.org/wiki/HTML_element) of a specific type, e.g. the second-level headers [h2](https://en.wikipedia.org/wiki/HTML_element#Basic_text)

elements specified by [attribute](https://en.wikipedia.org/wiki/HTML_attribute), in particular:

id: an identifier unique within the document

class: an identifier that can annotate multiple elements in a document

elements depending on how they are placed relative to others in the [document tree](https://en.wikipedia.org/wiki/Document_Object_Model).

Classes and IDs are case-sensitive, start with letters, and can include alphanumeric characters and underscores. A class may apply to any number of instances of any elements. An ID may only be applied to a single element.

Pseudo-classes are used in CSS selectors to permit formatting based on information that is not contained in the document tree. One example of a widely used pseudo-class is :hover, which identifies content only when the user "points to" the visible element, usually by holding the mouse cursor over it. It is appended to a selector as in a:hover or #elementid:hover. A pseudo-class classifies document elements, such as :link or :visited, whereas a pseudo-element makes a selection that may consist of partial elements, such as ::first-line or ::first-letter.[[5]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-5)

Selectors may be combined in many ways to achieve great specificity and flexibility.[[6]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-6) Multiple selectors may be joined in a spaced list to specify elements by location, element type, id, class, or any combination thereof. The order of the selectors is important. For example, div .myClass {color: red;} applies to all elements of class myClass that are inside div elements, whereas .myClass div {color: red;} applies to all div elements that are in elements of class myClass.

CSS information can be provided from various sources. These sources can be the web browser, the user and the author. The information from the author can be further classified into inline, media type, importance, selector specificity, rule order, inheritance and property definition. CSS style information can be in a separate document or it can be embedded into an HTML document. Multiple style sheets can be imported. Different styles can be applied depending on the output device being used; for example, the screen version can be quite different from the printed version, so that authors can tailor the presentation appropriately for each medium.

The style sheet with the highest priority controls the content display. Declarations not set in the highest priority source are passed on to a source of lower priority, such as the user agent style. This process is called cascading.

One of the goals of CSS is to allow users greater control over presentation. Someone who finds red italic headings difficult to read may apply a different style sheet. Depending on the browser and the web site, a user may choose from various style sheets provided by the designers, or may remove all added styles and view the site using the browser's default styling, or may override just the red italic heading style without altering other attributes.

CSS was first proposed by [Håkon Wium Lie](https://en.wikipedia.org/wiki/H%C3%A5kon_Wium_Lie) on October 10, 1994.[[16]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-chss-proposal-16) At the time, Lie was working with [Tim Berners-Lee](https://en.wikipedia.org/wiki/Tim_Berners-Lee) at [CERN](https://en.wikipedia.org/wiki/CERN).[[17]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-chapter20-17) Several other style sheet languages for the web were proposed around the same time, and discussions on public mailing lists and inside [World Wide Web Consortium](https://en.wikipedia.org/wiki/World_Wide_Web_Consortium) resulted in the first W3C CSS Recommendation (CSS1)[[18]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-w3c-css1-18) being released in 1996. In particular, [Bert Bos](https://en.wikipedia.org/wiki/Bert_Bos)' proposal was influential; he became co-author of CSS1 and is regarded as co-creator of CSS.[[19]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-WWW3-19)

Style sheets have existed in one form or another since the beginnings of Standard Generalized Markup Language ([SGML](https://en.wikipedia.org/wiki/SGML)) in the 1980s, and CSS was developed to provide style sheets for the web.[[20]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-css-phd-20) One requirement for a web style sheet language was for style sheets to come from different sources on the web. Therefore, existing style sheet languages like [DSSSL](https://en.wikipedia.org/wiki/Document_Style_Semantics_and_Specification_Language) and [FOSI](https://en.wikipedia.org/wiki/Formatting_Output_Specification_Instance) were not suitable. CSS, on the other hand, let a document's style be influenced by multiple style sheets by way of "cascading" styles.[[20]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-css-phd-20)

As [HTML](https://en.wikipedia.org/wiki/HTML) grew, it came to encompass a wider variety of stylistic capabilities to meet the demands of [web developers](https://en.wikipedia.org/wiki/Web_development). This evolution gave the designer more control over site appearance, at the cost of more complex HTML. Variations in [web browser](https://en.wikipedia.org/wiki/Web_browser) implementations, such as [ViolaWWW](https://en.wikipedia.org/wiki/ViolaWWW) and [WorldWideWeb](https://en.wikipedia.org/wiki/WorldWideWeb),[[21]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-IEEE-21) made consistent site appearance difficult, and users had less control over how web content was displayed. The browser/editor developed by Tim Berners-Lee had style sheets that were hard-coded into the program. The style sheets could therefore not be linked to documents on the web.[[22]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-cssdftw-22)[Robert Cailliau](https://en.wikipedia.org/wiki/Robert_Cailliau), also of CERN, wanted to separate the structure from the presentation so that different style sheets could describe different presentation for printing, screen-based presentations, and editors.[[21]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-IEEE-21)

Improving web presentation capabilities was a topic of interest to many in the web community and nine different style sheet languages were proposed on the www-style mailing list.[[20]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-css-phd-20) Of these nine proposals, two were especially influential on what became CSS: Cascading HTML Style Sheets[[16]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-chss-proposal-16) and Stream-based Style Sheet Proposal (SSP).[[19]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-WWW3-19)[[23]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-ssp-23) Two browsers served as testbeds for the initial proposals; Lie worked with [Yves Lafon](https://en.wikipedia.org/w/index.php?title=Yves_Lafon&action=edit&redlink=1) to implement CSS in [Dave Raggett](https://en.wikipedia.org/wiki/Dave_Raggett)'s [Arena](https://en.wikipedia.org/wiki/Arena_%28web_browser%29) browser.[[24]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-24)[[25]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-25)[[26]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-26)[Bert Bos](https://en.wikipedia.org/wiki/Bert_Bos) implemented his own SSP proposal in the [Argo](https://en.wikipedia.org/wiki/Argo_%28web_browser%29) browser.[[19]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-WWW3-19) Thereafter, Lie and Bos worked together to develop the CSS standard (the 'H' was removed from the name because these style sheets could also be applied to other markup languages besides HTML).[[17]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-chapter20-17)

Lie's proposal was presented at the "[Mosaic and the Web](https://en.wikipedia.org/w/index.php?title=Mosaic_and_the_Web&action=edit&redlink=1)" conference (later called WWW2) in [Chicago, Illinois](https://en.wikipedia.org/wiki/Chicago,_Illinois) in 1994, and again with Bert Bos in 1995.[[17]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-chapter20-17) Around this time the W3C was already being established, and took an interest in the development of CSS. It organized a workshop toward that end chaired by [Steven Pemberton](https://en.wikipedia.org/wiki/Steven_Pemberton). This resulted in W3C adding work on CSS to the deliverables of the HTML editorial review board (ERB). Lie and Bos were the primary technical staff on this aspect of the project, with additional members, including [Thomas Reardon](https://en.wikipedia.org/wiki/Thomas_Reardon) of [Microsoft](https://en.wikipedia.org/wiki/Microsoft), participating as well. In August 1996 [Netscape Communication Corporation](https://en.wikipedia.org/wiki/Netscape) presented an alternative style sheet language called [JavaScript Style Sheets](https://en.wikipedia.org/wiki/JavaScript_Style_Sheets) (JSSS).[[17]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-chapter20-17) The spec was never finished and is deprecated.[[27]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-27) By the end of 1996, CSS was ready to become official, and the CSS level 1 Recommendation was published in December.

Development of HTML, CSS, and the [DOM](https://en.wikipedia.org/wiki/Document_Object_Model) had all been taking place in one group, the HTML Editorial Review Board (ERB). Early in 1997, the ERB was split into three [working groups](https://en.wikipedia.org/wiki/Working_group): [HTML Working group](https://en.wikipedia.org/wiki/HTML_Working_group), chaired by [Dan Connolly](https://en.wikipedia.org/wiki/Dan_Connolly_%28computer_scientist%29) of W3C; DOM Working group, chaired by Lauren Wood of [SoftQuad](https://en.wikipedia.org/wiki/SoftQuad); and [CSS Working group](https://en.wikipedia.org/wiki/CSS_Working_group), chaired by [Chris Lilley](https://en.wikipedia.org/wiki/Chris_Lilley_%28W3C%29) of W3C.

The CSS Working Group began tackling issues that had not been addressed with CSS level 1, resulting in the creation of CSS level 2 on November 4, 1997. It was published as a W3C Recommendation on May 12, 1998. CSS level 3, which was started in 1998, is still under development as of 2014.

In 2005 the CSS Working Groups decided to enforce the requirements for standards more strictly. This meant that already published standards like CSS 2.1, CSS 3 Selectors and CSS 3 Text were pulled back from Candidate Recommendation to Working Draft level.

## 6.3 MYSQL Server

MySQL  is an [open-source](https://en.wikipedia.org/wiki/Open-source) [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS);[[6]](https://en.wikipedia.org/wiki/MySQL#cite_note-6) in July 2013, it was the world's second most widely used RDBMS, and the most widely used open-source [client–server model](https://en.wikipedia.org/wiki/Client%E2%80%93server_model) RDBMS. It is named after co-founder [Michael Widenius](https://en.wikipedia.org/wiki/Michael_Widenius)'s daughter, My. The [SQL](https://en.wikipedia.org/wiki/SQL) acronym stands for [Structured Query Language](https://en.wikipedia.org/wiki/Structured_Query_Language). The MySQL development project has made its [source code](https://en.wikipedia.org/wiki/Source_code) available under the terms of the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License), as well as under a variety of [proprietary](https://en.wikipedia.org/wiki/Proprietary_software) agreements. MySQL was owned and sponsored by a single [for-profit](https://en.wikipedia.org/wiki/Business) firm, the [Swedish](https://en.wikipedia.org/wiki/Sweden" \o "Sweden)company [MySQL AB](https://en.wikipedia.org/wiki/MySQL_AB), now owned by [Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation). For proprietary use, several paid editions are available, and offer additional functionality.

SQL Server Management Studio (SSMS) is a software application first launched with [Microsoft](https://en.wikipedia.org/wiki/Microsoft)[SQL Server 2005](https://en.wikipedia.org/wiki/Microsoft_SQL_Server) that is used for configuring, managing, and administering all components within [Microsoft SQL Server](https://en.wikipedia.org/wiki/Microsoft_SQL_Server). The tool includes both script editors and graphical tools which work with objects and features of the server.[[1]](https://en.wikipedia.org/wiki/SQL_Server_Management_Studio#cite_note-1)

A central feature of SSMS is the Object Explorer, which allows the user to browse, select, and act upon any of the objects within the server.[[2]](https://en.wikipedia.org/wiki/SQL_Server_Management_Studio#cite_note-2) It also shipped a separate Express edition that could be freely downloaded, however recent versions of SSMS are fully capable of connecting to and manage any SQL Server Express instance. Microsoft also incorporated backwards compatibility for older versions of SQL Server thus allowing a newer version of SSMS to connect to older versions of SQL Server instances.

Starting from version 11, the application was based on the [Visual Studio 2010](https://en.wikipedia.org/wiki/Visual_Studio_2010) shell, using [WPF](https://en.wikipedia.org/wiki/Windows_Presentation_Foundation) for the user interface.

In June 2015, Microsoft announced their intention to release future versions of SSMS independently of SQL Server database engine releases.[[3]](https://en.wikipedia.org/wiki/SQL_Server_Management_Studio#cite_note-3).

## 6.4PHP

PHP is a [server-side scripting](https://en.wikipedia.org/wiki/Server-side_scripting) language designed for [web development](https://en.wikipedia.org/wiki/Web_development) but also used as a [general-purpose programming language](https://en.wikipedia.org/wiki/General-purpose_programming_language). Originally created by [RasmusLerdorf](https://en.wikipedia.org/wiki/Rasmus_Lerdorf" \o "Rasmus Lerdorf) in 1994, the PHP [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) is now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for the[recursive](https://en.wikipedia.org/wiki/Recursive_acronym) [backronym](https://en.wikipedia.org/wiki/Backronym) PHP: Hypertext Preprocessor.

PHP code may be embedded into [HTML](https://en.wikipedia.org/wiki/HTML) code, or it can be used in combination with various [web template systems](https://en.wikipedia.org/wiki/Web_template_system), web content management system and [web frameworks](https://en.wikipedia.org/wiki/Web_framework). PHP code is usually processed by a PHP[interpreter](https://en.wikipedia.org/wiki/Interpreter_(computing)) implemented as a [module](https://en.wikipedia.org/wiki/Plugin_(computing)) in the web server or as a [Common Gateway Interface](https://en.wikipedia.org/wiki/Common_Gateway_Interface) (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a [command-line interface](https://en.wikipedia.org/wiki/Command-line_interface)(CLI) and can be used to implement [standalone](https://en.wikipedia.org/wiki/Computer_software) [graphical applications](https://en.wikipedia.org/wiki/Graphical_user_interface).

The standard PHP interpreter, powered by the [Zend Engine](https://en.wikipedia.org/wiki/Zend_Engine), is [free software](https://en.wikipedia.org/wiki/Free_software) released under the [PHP License](https://en.wikipedia.org/wiki/PHP_License). PHP has been widely ported and can be deployed on most web servers on almost every [operating system](https://en.wikipedia.org/wiki/Operating_system) and[platform](https://en.wikipedia.org/wiki/Computing_platform), free of charge.

The PHP language evolved without a written [formal specification](https://en.wikipedia.org/wiki/Formal_specification) or standard until 2014, leaving the canonical PHP interpreter as a [de facto](https://en.wikipedia.org/wiki/De_facto) standard. Since 2014 work has gone on to create a formal PHP specification.

PHP is a [server-side scripting](https://en.wikipedia.org/wiki/Server-side_scripting) language designed primarily for [web development](https://en.wikipedia.org/wiki/Web_development) but also used as a [general-purpose programming language](https://en.wikipedia.org/wiki/General-purpose_programming_language). Originally created by [Rasmus Lerdorf](https://en.wikipedia.org/wiki/Rasmus_Lerdorf) in 1994,[[4]](https://en.wikipedia.org/wiki/PHP#cite_note-History_of_PHP-4) the PHP [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) is now produced by The PHP Development Team.[[5]](https://en.wikipedia.org/wiki/PHP#cite_note-about_PHP-5) PHP originally stood for Personal Home Page,[[4]](https://en.wikipedia.org/wiki/PHP#cite_note-History_of_PHP-4) but it now stands for the [recursive acronym](https://en.wikipedia.org/wiki/Recursive_acronym) PHP: Hypertext Preprocessor.[[6]](https://en.wikipedia.org/wiki/PHP#cite_note-6)

PHP code may be embedded into [HTML](https://en.wikipedia.org/wiki/HTML) or HTML5 code, or it can be used in combination with various [web template systems](https://en.wikipedia.org/wiki/Web_template_system), [web content management systems](https://en.wikipedia.org/wiki/Web_content_management_system) and [web frameworks](https://en.wikipedia.org/wiki/Web_framework). PHP code is usually processed by a PHP [interpreter](https://en.wikipedia.org/wiki/Interpreter_%28computing%29) implemented as a [module](https://en.wikipedia.org/wiki/Plugin_%28computing%29) in the web server or as a [Common Gateway Interface](https://en.wikipedia.org/wiki/Common_Gateway_Interface) (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a [command-line interface](https://en.wikipedia.org/wiki/Command-line_interface) (CLI) and can be used to implement [standalone](https://en.wikipedia.org/wiki/Computer_software)[graphical applications](https://en.wikipedia.org/wiki/Graphical_user_interface).[[7]](https://en.wikipedia.org/wiki/PHP#cite_note-7)

The standard PHP interpreter, powered by the [Zend Engine](https://en.wikipedia.org/wiki/Zend_Engine), is [free software](https://en.wikipedia.org/wiki/Free_software) released under the [PHP License](https://en.wikipedia.org/wiki/PHP_License). PHP has been widely ported and can be deployed on most web servers on almost every [operating system](https://en.wikipedia.org/wiki/Operating_system) and [platform](https://en.wikipedia.org/wiki/Computing_platform), free of charge.[[8]](https://en.wikipedia.org/wiki/PHP#cite_note-foundations-8)

The PHP language evolved without a written [formal specification](https://en.wikipedia.org/wiki/Formal_specification) or standard until 2014, leaving the canonical PHP interpreter as a [de facto](https://en.wikipedia.org/wiki/De_facto) standard. Since 2014 work has gone on to create a formal PHP specification.[[9]](https://en.wikipedia.org/wiki/PHP#cite_note-9)

PHP development began in 1995 when [Rasmus Lerdorf](https://en.wikipedia.org/wiki/Rasmus_Lerdorf) wrote several [Common Gateway Interface](https://en.wikipedia.org/wiki/Common_Gateway_Interface) (CGI) programs in C,[[10]](https://en.wikipedia.org/wiki/PHP#cite_note-php_origins-10)[[11]](https://en.wikipedia.org/wiki/PHP#cite_note-11)[[12]](https://en.wikipedia.org/wiki/PHP#cite_note-12) which he used to maintain his [personal homepage](https://en.wikipedia.org/wiki/Personal_homepage). He extended them to work with [web forms](https://en.wikipedia.org/wiki/Web_form) and to communicate with [databases](https://en.wikipedia.org/wiki/Database), and called this implementation "Personal Home Page/Forms Interpreter" or PHP/FI.

PHP/FI could help to build simple, dynamic [web applications](https://en.wikipedia.org/wiki/Web_application). To accelerate [bug](https://en.wikipedia.org/wiki/Software_bug) reporting and to improve the code, Lerdorf initially announced the release of PHP/FI as "Personal Home Page Tools (PHP Tools) version 1.0" on the [Usenet](https://en.wikipedia.org/wiki/Usenet)discussion group comp.infosystems.www.authoring.cgi on June 8, 1995.[[13]](https://en.wikipedia.org/wiki/PHP#cite_note-13)[[14]](https://en.wikipedia.org/wiki/PHP#cite_note-14) This release already had the basic functionality that PHP has as of 2013. This included [Perl-like variables](https://en.wikipedia.org/wiki/Local_variable#Local_variables_in_Perl), form handling, and the ability to embed HTML. The [syntax](https://en.wikipedia.org/wiki/Syntax) resembled that of Perl but was simpler, more limited and less consistent.[[5]](https://en.wikipedia.org/wiki/PHP#cite_note-about_PHP-5)

Lerdorf did not intend the early PHP to become a new programming language, but it grew organically, with Lerdorf noting in retrospect: "I don’t know how to stop it, there was never any intent to write a programming language […] I have absolutely no idea how to write a programming language, I just kept adding the next logical step on the way."[[15]](https://en.wikipedia.org/wiki/PHP#cite_note-itconversations-15) A development team began to form and, after months of work and [beta](https://en.wikipedia.org/wiki/Beta_development_stage) testing, officially released PHP/FI 2 in November 1997.

The fact that PHP lacked an original overall design but instead developed organically has led to inconsistent naming of functions and inconsistent ordering of their parameters.[[16]](https://en.wikipedia.org/wiki/PHP#cite_note-16) In some cases, the function names were chosen to match the lower-level libraries which PHP was "wrapping",[[17]](https://en.wikipedia.org/wiki/PHP#cite_note-17) while in some very early versions of PHP the length of the function names was used internally as a [hash function](https://en.wikipedia.org/wiki/Hash_function), so names were chosen to improve the distribution of hash values.[[18]](https://en.wikipedia.org/wiki/PHP#cite_note-php-internals-70691-18)

## 6.5 ANGULAR JAVA SCRIPT

AngularJS (commonly referred to as "Angular" or "Angular.js") is an [open-source](https://en.wikipedia.org/wiki/Open-source_software) [web application framework](https://en.wikipedia.org/wiki/Web_application_framework) mainly maintained by [Google](https://en.wikipedia.org/wiki/Google) and by a community of individuals and corporations to address many of the challenges encountered in developing [single-page applications](https://en.wikipedia.org/wiki/Single-page_application). It aims to simplify both the development and the [testing](https://en.wikipedia.org/wiki/Software_testing) of such applications by providing a framework for client-side [model–view–controller](https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller) (MVC) and [model–view–viewmodel](https://en.wikipedia.org/wiki/Model_View_ViewModel)(MVVM) architectures, along with components commonly used in [rich Internet applications](https://en.wikipedia.org/wiki/Rich_Internet_Application).

The AngularJS framework works by first reading the [HTML](https://en.wikipedia.org/wiki/HTML) page, which has embedded into it additional custom [tag attributes](https://en.wikipedia.org/wiki/HTML_attribute). Angular interprets those attributes as directives to bind input or output parts of the page to a model that is represented by standard [JavaScript](https://en.wikipedia.org/wiki/JavaScript) variables. The values of those JavaScript variables can be manually set within the code, or retrieved from static or dynamic [JSON](https://en.wikipedia.org/wiki/JSON) resources.

According to [JavaScript](https://en.wikipedia.org/wiki/JavaScript) analytics service [Libscore](https://en.wikipedia.org/wiki/Libscore" \o "Libscore), AngularJS is used on the websites of [Wolfram Alpha](https://en.wikipedia.org/wiki/Wolfram_Alpha), [NBC](https://en.wikipedia.org/wiki/NBC" \o "NBC),[Walgreens](https://en.wikipedia.org/wiki/Walgreens), [Intel](https://en.wikipedia.org/wiki/Intel), [Sprint](https://en.wikipedia.org/wiki/Sprint_Nextel), [ABC News](https://en.wikipedia.org/wiki/ABC_News), and approximately 8,400 other sites out of 1 million tested in July 2015.

AngularJS is the frontend part of the [MEAN stack](https://en.wikipedia.org/wiki/MEAN_(software_bundle)), consisting of [MongoDB](https://en.wikipedia.org/wiki/MongoDB) database, [Express.js](https://en.wikipedia.org/wiki/Express.js) web application server framework, Angular.js itself, and [Node.js](https://en.wikipedia.org/wiki/Node.js) runtime environment.

AngularJS is an open source web application framework. It was originally developed in 2009 by Misko Hevery and Adam Abrons. It is now maintained by Google. Its latest version is 1.4.3.

Definition of AngularJS as put by its [official documentation](https://docs.angularjs.org/guide/introduction) is as follows −

AngularJS is a structural framework for dynamic web apps. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application's components clearly and succinctly. Angular's data binding and dependency injection eliminate much of the code you currently have to write. And it all happens within the browser, making it an ideal partner with any server technology.

**Features**

* AngularJS is a powerful JavaScript based development framework to create RICH Internet Application(RIA).
* AngularJS provides developers options to write client side application (using JavaScript) in a clean MVC(Model View Controller) way.
* Application written in AngularJS is cross-browser compliant. AngularJS automatically handles JavaScript code suitable for each browser.
* AngularJS is open source, completely free, and used by thousands of developers around the world. It is licensed under the Apache License version 2.0.
* Overall, AngularJS is a framework to build large scale and high performance web application while keeping them as easy-to-maintain.

**Core Features**

Following are most important core features of AngularJS −

* **Data-binding** − It is the automatic synchronization of data between model and view components.
* **Scope −** These are objects that refer to the model. They act as a glue between controller and view.
* **Controller −** These are JavaScript functions that are bound to a particular scope.
* **Services −** AngularJS come with several built-in services for example $https: to make a XMLHttpRequests. These are singleton objects which are instantiated only once in app.
* **Filters** − These select a subset of items from an array and returns a new array.
* **Directives −** Directives are markers on DOM elements (such as elements, attributes, css, and more). These can be used to create custom HTML tags that serve as new, custom widgets. AngularJS has built-in directives (ngBind, ngModel...)
* **Templates** − These are the rendered view with information from the controller and model. These can be a single file (like index.html) or multiple views in one page using "partials".
* **Routing** − It is concept of switching views.
* **Model View Whatever** − MVC is a design pattern for dividing an application into different parts (called Model, View and Controller), each with distinct responsibilities. AngularJS does not implement MVC in the traditional sense, but rather something closer to MVVM (Model-View-ViewModel). The Angular JS team refers it humorously as Model View Whatever.
* **Deep Linking** − Deep linking allows you to encode the state of application in the URL so that it can be bookmarked. The application can then be restored from the URL to the same state.
* **Dependency Injection** − AngularJS has a built-in dependency injection subsystem that helps the developer by making the application easier to develop, understand, and test.

**CHAPTER 7- SYSTEM STUDY**

**7.1 FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

# CHAPTER 8-TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**8.1Non Functional Requirements**

Non-functional requirements are the quality requirements that stipulate how well software does what it has to do. These are Quality attributes of any system; these can be seen at the execution of the system and they can also be the part of the system architecture.

**8.2 Accuracy:**

The system will be accurate and reliable based on the design architecture. If there is any problem in the accuracy then the system will provide alternative ways to solve the problem.

**8.3 Usability:**

The proposed system will be simple and easy to use by the users. The users will comfort in order to communicate with the system. The user will be provided with an easy interface of the system.

**8.4 Accessibility:**

The system will be accessible through internet and there should be no any known problem.

* 1. **Performance:**

The system performance will be at its best when performing the functionality of the system.

* 1. **Reliability:**

The proposed system will be reliable in all circumstances and if there is any problem that will be affectively handle in the design.

* 1. **Security:**

The proposed system will be highly secured; every user will be required registration and username/password to use the system. The system will do the proper authorization and authentication of the users based on their types and their requirements. The proposed system will be designed persistently to avoid any misuse of the application.

# CHAPTER 9-SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**TESTCASE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO** | **SCENARIO** | **INPUT** | **EXPECTED**  **OUTPUT** | **ACTUAL OUTPUT** |
| 1 | Patient register | Username and Password | Patient details register successfully | Register successfully  Or  Register unsuccessfully |
| 2 | Patient Login | Email and Password | If correct directed to home page otherwise show “Login Successfully” | Login successfully or  Login unsuccessfully |
| 3 | List of Doctor | Patient check and view Doctor details | Patient view Doctor details | View all Doctor |
| 4 | Book Doctor | Patient Book the Doctor | If all the doctor correct Patient book the “Booked successfully” | Booked successfully or  Not Booked |
| 5 | View Medical Report and Bills | Patient view Medical Report and Bills | View Medical Report and Bills | Show Medical Report and Bills |
| 6 | List of Medical Report | Patient check and view Medical Report | Patient view Medical Report | View all Medical Report |
| 7 | Medical Prescription | Patient check and view all Prescription | Patient view Prescription | View division Prescription |
| 8 | Medical Information | Patient check and view all Information | Patient view Information | View division Information |
| 9 | Medical History | Patient check and view all History | Patient view History | View division History |
| 10 | Smart Health Card | Patient add the Health Card | all the Health Card “created successfully” | created successfully or  unsuccessfully |
| 11 | My Profile | Patient can edit and profile | If any changes or user can edit profile | Updated successfully or unsuccessfully |
| 12 | Doctor register | Username and Password | Doctor details register successfully | Register successfully  Or  Register unsuccessfully |
| 13 | Doctor Login | Email and Password | If correct directed to home page otherwise show “Login Successfully” | Login successfully or  Login unsuccessfully |
| 14 | Generate QR code | Doctor add the QR code | all the QR code “created successfully” | created successfully or  unsuccessfully |
| 15 | Scan QR code | Doctor add the QR code | all the Doctor details “Scanned  successfully” | Scanned successfully or  unsuccessfully |
| 16 | Create Hospital | Hospital created by all details like name,address, phone etc | All the details Create successfully | Created successfully or created unsuccessfully |
| 17 | Update Details | Doctor add the details | all the details “updated  successfully” | updated successfully or  unsuccessfully |
| 18 | Booking Details | Doctor check and view all Booking Details | Doctor view Booking Details | View division Booking Details |
| 20 | Update Medical Report and Bills | Doctor add the Medical Report and Bills details | all the details “updated  successfully” | updated successfully or  unsuccessfully |
| 21 | Update Medical Prescription | Doctor add the Medical Prescription | all the details “updated  successfully” | updated successfully or  unsuccessfully |
| 22 | Update Medical Information | Doctor add the Medical Information | all the details “updated  successfully” | updated successfully or  unsuccessfully |
| 23 | Update Medical History | Doctor add the Medical History | all the details “updated  successfully” | updated successfully or  unsuccessfully |
| 24 | View Health Card | Doctor view Health Card | View Health Card | Show Health Card |

**TYPES OF TESTS**

**Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**9.1 Unit Testing:**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page

**9.2 Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**9.3 Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

# 

# CHAPTER 10– CONCLUSIONS

In our country, the lack of patient prior medical information at a particular instance of time may be leading to medical errors and patient death. So, to reduce the medical errors it has been decided to implement an application with a personal smart health card for every user with his/her medical information. The user has to maintain all his/her personal and medical data in the cloud database. In any emergency, the person has to be identified very quickly and he/she needs immediate medical attention by accessing previous medical information by scanning QR code which is printed on the smart card. This approach will be flexible and efficient for all users, clinics, and pharmacies because the user has just to show his/her smart card to thproviders. Finally, the tips and suggestions from the doctor are notified to the patient account after treatment. It is always required to enhance this proposed model with physical cards and reminders for the prescribed medicines in future.

# CHAPTER 11- REFERENCES

* [1]. Jacek Ruminski, Krzysztof Czuszynski, ”Interaction wit h medical dat a using QR-codes”, Human System Int eractions (HSI), June 2014
* [2]. Paschou Mersini, Evangelos Sakkopoulos, Athanasios Tsakalidis, ”APPification of Hospital Healthcare and Data Management using QR codes”, Informat ion, Int elligence, Systems and Applicat ions (IISA), July 2013
* [3]. Chalumuru Suresh,P . Geet hika Choudary, Itabada Pranav Souri, Bolla Vineela, “Health St udio - An Web Applicat ion for Self Assessment”, Int ernat ional Conference on Communicat ion and Elect ronics Syst ems (ICCES), February 2020
* [4]. Sudha.G, Ganesan.R, ”Secure Transmission Medical Dat a for P ervasive Healthcare Syst em using Web”, International Conference on Communicat ions and Signal P rocessing (ICCSP ), April 2013
* [5]. Mungyu Bae, Suk Kyu Lee, Seungho Yoo, and Hwangnam Kim,” FASE: Fast Authentication System for E-Health”, International Conference on Ubiquit ous and Fut ure Net works (ICUFN), July 2013
* [6]Mr. Mahale Kishor M, Prof. Dhomse Kanchan B, ”Study of machine learning algorithms for special disease predict ion using principal of component analysis”, International Conference on Global Trends in Signal Processing, Informat ion Comput ing and Communication (ICGTSP ICC), December 2016
* [7] Chayakrit Charoensiriwath, Navaporn Surasvadi, Suporn Pongnumkul, Thunyasit P holprasit, “Applying QR Code and Mobile Application to Improve Service Process in Thai Hospital”, Int ernat ional Joint Conference on Comput er Science and Soft ware Engineering (IJCSSE), July 2015
* [8] Rizal Mohd Nor, “ScanMed: A Mobile Medicine Adherence Applicat ion with Intake Validation Using QR Code”, Information and Communication Technology for t he Muslim World (ICT4M), November 2016
* [9] Won-Jae Yi, Jafar Saniie, “Patient Centered Real-Time Mobile Health Monitoring System”, E-Health Telecommunicat ion syst ems and Net works, January 2016
* Rameswari. R, Divya. N, “Smart Health Care Monitoring Syst em Using Web Application”, IJRT E, November 2018.

# CHAPTER 12 – Coding

<!DOCTYPE html>

<html ng-app="myapp">

<head>

<!-- Basic -->

<meta charset="utf-8">

<meta name="keywords" content="HTML5 Template" />

<meta name="description" >

<meta name="author" content="pixelgeeklab.com">

<!-- Mobile Metas -->

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!-- Web Fonts -->

<link href='css/css.css' rel='stylesheet' type='text/css'>

<link rel="stylesheet" href="fonts/font.css">

<!-- Vendor CSS -->

<link rel="stylesheet" href="vendor/fontawesome/css/font-awesome.css">

<link rel="stylesheet" href="vendor/owlcarousel/owl.carousel.css" media="screen">

<link rel="stylesheet" href="vendor/owlcarousel/owl.theme.css" media="screen">

<link href="vendor/owl-carousel/owl.transitions.html" rel="stylesheet" media="screen">

<link rel="stylesheet" href="vendor/flexslider/flexslider.css" media="screen">

<link rel="stylesheet" href="vendor/chosen/chosen.css" media="screen">

<link rel="stylesheet" href="vendor/magnific-popup/magnific-popup.css" media="screen">

<!-- Theme CSS -->

<link rel="stylesheet" href="css/theme.css">

<link rel="stylesheet" href="css/theme-animate.css">

<!-- Style Switcher-->

<!-- Head libs -->

<script src="vendor/modernizr/modernizr.js"></script>

<!--[if IE]>

<link rel="stylesheet" href="css/ie.css">

<![endif]-->

<!--[if lte IE 8]>

<script src="vendor/respond/respond.js"></script>

<script src="vendor/excanvas/excanvas.js"></script>

<![endif]-->

<style>

.page-wrap {

max-width: 75rem;

margin: 0 auto;

}

h1 {

font-size: 1.5rem;

letter-spacing: -1px;

margin: 1.25rem 0;

}

input:focus ~ label, textarea:focus ~ label, input:valid ~ label, textarea:valid ~ label {

font-size: 0.75em;

top: -2.25rem;

-webkit-transition: all 0.125s ease;

transition: all 0.125s ease;

}

.styled-input {

left:10px;

width: 90%;

margin: 2rem 0 1rem;

position: relative;

}

.styled-input label {

color: #999;

padding: 1rem;

position: absolute;

top: 0;

left: 0;

-webkit-transition: all 0.25s ease;

transition: all 0.25s ease;

pointer-events: none;

}

input,

textarea {

position: relative;

height: 2.9em;

border: 0px solid #bebebe;

border-bottom-width: 1px;

width: 100%;

}

input ~ span,

textarea ~ span {

display: block;

width: 0;

height:1px;

background: #03a3e8;

position: absolute;

bottom: 0;

left: 0;

-webkit-transition: all 0.125s ease;

transition: all 0.125s ease;

}

input:focus,

textarea:focus { outline: 0; }

input:focus ~ span,

textarea:focus ~ span {

width: 100%;

-webkit-transition: all 0.075s ease;

transition: all 0.075s ease;

}

textarea {

width: 100%;

min-height: 15em;

}

</style>

<!-- script back button -->

<script src="cordova.js"></script>

<script>

function onLoad()

{

document.addEventListener("deviceready", deviceReady, false);

}

function deviceReady()

{

document.addEventListener("backbutton", backButtonCallback, false);

}

function backButtonCallback()

{

navigator.app.exitApp();

}

</script>

</head>

<!-- angular js -->

<script src="js/angular-1.3.js"></script>

<script src="js/angular\_cookies.js"></script>

<body ng-controller="myappCtrl" class="front bg-pattern-dark">

<div class="body">

<header id="header">

<div class="header-top" style="height:8em;">

<div class="container">

<a href="#offcanvas" class="uk-navbar-toggle" style="float:left; margin:50px 5px 10px 5px;" data-uk-offcanvas>

<span class="sr-only">Toggle navigation</span>

<i class="fa fa-bars"></i>

</a>

<nav>

<ul class="nav nav-pills nav-top nav-top-left">

<div class="head">Garbage Management </div>

</ul>

<ul class="nav nav-pills nav-top nav-top-right">

<li class="login">

<a data-target=".form-wrapper" data-toggle="modal" ng-click="admin\_logout()"><img style="margin-top:37px;" class="fa icon-x" src="images/icons/user1.png"/> </a>

</li>

</ul>

</nav>

</div>

</div>

</header>

</div>

<div id="offcanvas" class="uk-offcanvas">

<div class="uk-offcanvas-bar">

<ul id="main-menu-offcanvas" class="uk-nav-offcanvas" data-uk-nav>

<h2 style="color:#000;padding:1.5em;font-weight:bold;font-family:'Quicksand';">Garbage Management</h2>

<span class="uk-parent">

<a href="user\_home.html"><img src="images/icons/house.png" class="fa icon"> &nbsp;&nbsp;&nbsp;Home</a>

</span>

<span class="uk-parent">

<a href="#"><img src="images/icons/laptop-1.png" class="fa icon"> &nbsp;&nbsp;&nbsp;About Us</a>

</span>

<span class="uk-parent">

<a href="#"><img src="images/icons/smartphone.png" class="fa icon"> &nbsp;&nbsp;&nbsp;Contact Us</a>

</span>

<span class="uk-parent">

<a href="#"><img src="images/icons/notepad.png" class="fa icon"> &nbsp;Terms & Conditions</a>

</span>

<span class="uk-parent">

<a href="#"><img src="images/icons/speech-bubble.png" class="fa icon"> &nbsp;&nbsp;&nbsp;Versions</a>

</span>

<span class="uk-parent">

<a style="cursor:pointer" ng-click="user\_logout()" >

<img src="images/icons/smartphone-2.png" class="fa icon"> &nbsp;&nbsp;&nbsp;Logout</a>

</span>

</ul>

</div>

</div>

<div class="content" style="margin-top:3em">

<div class="container">

<section class="content-inner">

<div class="container-fluid">

<div class="col-md-4" ng-repeat="x in balance" >

<a ng-bind="x.balance"></a>

</div>

<div class="row" style="margin-left:50px;">

<div class="col-md-6" >

<img style="width:80px; height:80px;" src="images/garbage.png"/>

<a href="view\_business.html" style="font-family:'Quicksand';border-radius:2px;box-shadow:1px 1px 1px 1.5px #B6B6B4; margin:1em 0 3em 1em; font-weight:bold;color:rgb(55,71,79);font-size:1.11em;padding:0.3em">

Register Complaints</a>

</div>

<div class="col-md-6" >

<img style="width:80px; height:80px;" src="images/garbage.png"/>

<a href="view\_user\_complaint.html" style="font-family:'Quicksand';border-radius:2px;box-shadow:1px 1px 1px 1.5px #B6B6B4; margin:1em 0 3em 1em; font-weight:bold;color:rgb(55,71,79);font-size:1.11em;padding:0.3em">

My Complaints</a>

</div>

<!--

<div class="col-md-6" >

<img style="width:80px; height:80px;" src="images/learning.png"/>

<a href="view\_students.html" style="border-radius:2px;box-shadow:1px 1px 1px 1.5px #B6B6B4; margin:1em 0 3em 1em; font-weight:bold;color:rgb(55,71,79);font-size:1.11em;padding:0.3em">

Update Business</a>

</div>

-->

<div class="col-md-6" >

<img style="width:80px; height:80px;" src="images/user.png"/>

<a href="user\_update\_info.html" style="border-radius:2px;box-shadow:1px 1px 1px 1.5px #B6B6B4; margin:1em 0 3em 1em; font-weight:bold;color:rgb(55,71,79);font-size:1.11em;padding:0.3em">

My Profile</a>

</div>

</div>

</div>

</section>

<!--<h2 class="content-sub-heading" style="text-align:center;color:rgb(69,90,100);font-family:'Quicksand';">

Analysis Report</h2>

<div class="row" style="margin-left:50px;">

<div class="col-md-6" >

<h2 class="content-sub-heading" style="text-align:center;color:rgb(69,90,100);font-family:'Quicksand';">

Enquries</h2>

<table class="table table-hover table-stripe" title="Default Tabl">

<thead class="fontsforweb\_fontid\_1409" style="color:#000;">

<tr>

<th >No of Enquries</th>

<th >Estimated Cost</th>

</tr>

</thead>

<tbody ng-repeat="x in enquiry">

<tr>

<td ng-bind="x.total"></td>

<td ng-bind="x.estimate\_cost"></td>

</tr>

</tbody>

</table>

</div>

<div class="col-md-6" >

<h2 class="content-sub-heading" style="text-align:center;color:rgb(69,90,100);font-family:'Quicksand';">

Project's Ongoing</h2>

<table class="table table-hover table-stripe" title="Default Tabl">

<thead class="fontsforweb\_fontid\_1409" style="color:#000;">

<tr>

<th >No of Projects</th>

<th >Estimated Payment</th>

<th >Payed</th>

<th >Pending Payment</th>

</tr>

</thead>

<tbody ng-repeat="x in project ">

<tr>

<td ng-bind="x.total"></td>

<td ng-bind="x.estimate\_cost"></td>

<td ng-bind="x.payed"></td>

<td >{{x.estimate\_cost - x.payed}}</td>

</tr>

</tbody>

</table>

</div>

</div>-->

<!--

<h2 class="content-sub-heading" style="text-align:center;color:rgb(69,90,100);font-family:'Quicksand';">

Project Management</h2>

<!-- Search Results

<div style="position: relative;left: 10px;">

<!--<h2 class="content-sub-heading" style="color:rgb(69,90,100);font-family:'Quicksand';">Search Results</h2>

<form class="form" >

<div class="styled-input">

<input ng-model="search\_field" type="text" required />

<label>Search</label>

<span></span>

</div>

</form>

</div>

<div class="col-md-6" >

<table class="table table-hover table-stripe" title="Default Tabl">

<thead class="fontsforweb\_fontid\_1409" style="color:#000;">

<tr>

<th >Project</th>

<th >Student Name</th>

<th >Start Date</th>

<th >Delivery Date</th>

<th >% Completed</th>

<th >Mobile</th>

<th >Email</th>

</tr>

</thead>

<tbody ng-repeat="x in task ">

<tr>

<td ng-bind="x.lname"></td>

<td ng-bind="x.email"></td>

<td ng-bind="x.chequeno"></td>

<td ng-bind="x.bank"></td>

</tr>

</tbody>

</table>

</div> -->

</div>

</div>

<script src="vendor/jquery/jquery.js"></script>

<script src="vendor/bootstrap/bootstrap.js"></script>

<script src="vendor/jquery.validation/jquery.validation.js"></script>

<script src="vendor/owlcarousel/owl.carousel.js"></script>

<script src="vendor/flexslider/jquery.flexslider-min.js"></script>

<script src="vendor/countdown/countdown.min.js"></script>

<script src="vendor/chosen/chosen.jquery.min.js"></script>

<script src="vendor/pricefilter/jquery.pricefilter.js"></script>

<script src="vendor/masonry/imagesloaded.pkgd.min.js"></script>

<script src="vendor/masonry/masonry.pkgd.min.js"></script>

<script src="vendor/uikit/uikit.js"></script>

<script src="vendor/magnific-popup/jquery.magnific-popup.js"></script>

<!-- Theme Base, Components and Settings -->

<script src="js/theme.js"></script>

<!-- Style Switcher -->

<script type="text/javascript" src="style-switcher/js/switcher.js"></script>

<!-- angular js -->

<script src="js/angular\_data.js"></script>

<body onload="onLoad()">

</body>

</html>

<?php

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* CPanel \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*/

/\* Following code will match admin login credentials \*/

//user temp array

$response = array();

// include db connect class

require\_once \_\_DIR\_\_ . '/db\_connect.php';

// connecting to db

$db = new DB\_CONNECT();

$data = json\_decode(file\_get\_contents("php://input"));

//$get\_field\_3 = mysql\_real\_escape\_string($data->dept);

$result = mysql\_query("SELECT \* FROM garbage ");

if(mysql\_num\_rows($result))

{

$response["details"] = array();

while($Alldetails = mysql\_fetch\_array($result))

{

// temp user array

$details = array();

$details = $Alldetails;

array\_push($response["details"],$details);

}

$response["success"] = 1;

echo json\_encode($response);

}

else

{

$response["success"] = 0;

echo json\_encode($response);

}

?>

<?php

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* CPanel \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*/

/\* Following code will match admin login credentials \*/

//user temp array

$response = array();

// include db connect class

require\_once \_\_DIR\_\_ . '/db\_connect.php';

// connecting to db

$db = new DB\_CONNECT();

$data = json\_decode(file\_get\_contents("php://input"));

$get\_email = "48";//mysql\_real\_escape\_string($data->email);

$result = mysql\_query("SELECT \* FROM answer where question='$get\_email' ");

$get\_quality\_rating =0;

$get\_pol\_rating =0;

$get\_money\_rating =0;

$get\_dis\_rating =0;

$count =0;

if(mysql\_num\_rows($result))

{

$response["details"] = array();

while($Alldetails = mysql\_fetch\_array($result))

{

// temp user array

$details = array();

$details = $Alldetails;

$question = $Alldetails["field\_3"];

$get\_rating = $Alldetails["field\_9"];

// Find the age metrics

if (strcmp($question,'Quality of Service' )==0)

{

$get\_quality\_rating = $get\_quality\_rating + $get\_rating;

$count++;

}

if (strcmp($question,'Politeness' )==0)

{

$get\_pol\_rating = $get\_pol\_rating + $get\_rating;

}

if (strcmp($question,'Value for Money' )==0)

{

$get\_money\_rating = $get\_money\_rating + $get\_rating;

}

if (strcmp($question,'Discount Policy' )==0)

{

$get\_dis\_rating = $get\_dis\_rating + $get\_rating;

}

}

//echo $count;

$details["get\_dis\_rating"] = $get\_dis\_rating/$count;

$details["get\_money\_rating"] = $get\_money\_rating/$count;

$details["get\_pol\_rating"] = $get\_pol\_rating/$count;

$details["get\_quality\_rating"] = $get\_quality\_rating/$count;

array\_push($response["details"],$details);

$response["success"] = 1;

echo json\_encode($response);

}

else

{

$response["success"] = 0;

echo json\_encode($response);

}

?>