

1. Introduction

the analysis phase is largely based on the feasibility study. Rather it would not be wrong to say that the analysis and feasibility phases overlap. High-level analysis begins during the feasibility study. Though analysis is represented as one phase of the system development life cycle (SDLC), this is not true. Analysis begins with system initialization and continues until its maintenance.

Even after successful implementation of the system, analysis may play its role for periodic Human Body is a very complex and sophisticated structure and comprises of millions of functions. All these complicated functions have been understood by man him, partby-part their research and experiments. As science and technology progressed, medicine became an integral part of the research. Gradually, medical science became an entirely new branch of science. As of today, the Health Sector comprises of Medical institutions i.e.

Hospitals, HOSPITALs etc. research and development institutions and medical colleges. Thus the Health sector aims at providing the best medical facilities to the common man.

1. Problem Statement

Since Hospital is associated with the lives of common people and their day-to-day routines so I decided to work on this project. The manual handling of the record is time consuming and highly prone to error. The purpose of this project is to automate or make online, the process of day-today activities like Room activities, Admission of New Patient, Discharge of Patient, Assign a Doctor, and finally compute the bill etc. I have tried my best to make the complicated process Hospital Management System as simple as possible using Structured & Modular technique & Menu oriented interface. I have tried to design the software in such a way that user may not have any difficulty in using this package & further expansion is possible without much effort. Even though I cannot claim that this work to be entirely exhaustive, the main purpose of my exercise is perform each Hospital's activity in computerized way rather than manually which is time consuming. I am confident that this software package can be readily used by nonprogramming personal avoiding human handled chance of error.

2. Analysis

System Analysis is a separation of a substance into parts for study and their implementation and detailed examination. Before designing any system it is important that the nature of the business and the way it currently operates are clearly understood. The detailed examination provides the specific data required during designing in order to ensure that all the

client's requirements are fulfilled. The investigation or the study conducted during maintenance and up gradation of the system. One of the main causes of project failures is inadequate understanding, and one of the main causes of inadequate understanding of the requirements is the poor planning of system analysis.

1. EXISTING SYSTEM:

Hospitals currently use a manual system for the management and maintainance of critical information. The current system requires numerous paper forms, with data stores spread through out the hospital management infrastructure. Often information is incomplete or does not follow management standards. Forms are often lost in transit between departments requiring a comprehensive auditing process to ensure that no vital information is lost. Multiple copies of the same information exist in the hospital and may lead to inconsistencies in data in various data stores.

2. PROPOSED SYSTEM:

The Hospital Management System is designed for any hospital to replace their existing manual paper based system. The new system is to control the information of patients. Room availability, staff and operating room schedules and patient invoices. These services are to be provided in an efficient, cost effective manner, with the goal of reducing the time and resources currently required for such tasks .

3. FEASIBILITY STUDY

The feasibility of the project is analysed 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:

A) Economic Feasibility

This study is carried out to check the economic impact will have on 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 customised products have to be purchased.

B) 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 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 for the implementing this system.

C) Operational 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.

4. SOFTWARE SPECIFICATION

HTML:

HTML or Hypertext Markup Language is the standard markup language used to create web pages. HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent *empty elements* and so are unpaired, for example . The first tag in a pair is the *start tag*, and the second tag is the *end tag* (they are also called *opening tags* and *closing tags*). Though not always necessary, it is best practice to append a slash to tags which are not paired with a closing tag.

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language rather than a programming language. HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create

interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages.

A.) Objective of project

Hospital are the essential part of our lives, providing best medical facilities to people suffering from various ailments, which may be due to change in climatic conditions, increased work-load, emotional trauma stress etc. It is necessary for the hospitals to keep track of its day-to-day activities & records of its patients, doctors, nurses, ward boys and other staff personals that keep the hospital running smoothly & successfully. But keeping track of all the activities and their records on paper is very cumbersome and error prone. It also is very inefficient and a timeconsuming process Observing the continuous increase in population and number of people visiting the hospital. Recording and maintaining all these records is highly unreliable, inefficient and error-prone. It is also not economically & technically feasible to maintain these records on paper. Thus keeping the working of the manual system as the basis of our project. We have developed an automated version of the manual system, named as "Administration support system for medical institutions". The main aim of our project is to provide a paper-less hospital up to 90%. It also aims at providing low-cost reliable automation of the existing systems. The system also provides excellent security of data at every level of user-system interaction and also provides robust & reliable storage and backup facilities.

B.) Requirement Gathering

The software requirement engineering determines the functional or non-functional requirements for engineering software. The requirements engineering is the first stage of any software project development. It is the process of determining functions of the software systems. The process encompasses all activities concerned with the requirements eliciting, analyzing, documenting, validating and managing software or systems. In requirement engineering [1] the real world goals are explored and established for the software system that is being developed. Before any project, the requirements of the user are collected to accomplish the user's task. The first stage of requirements engineering process is requirement gathering. Unfortunately, complete requirements cannot be perceived at a given point of time. The reason is that they evolve with time mostly they are observed after the system deployment. This

C.) Hardware requirement

- i. A device (Computer/laptop)
- ii. Memory (RAM): Minimum 2GB RAM iii. Processor: Minimum 1GHZ; Recommended 2GHZ or more.
- iv. Hard disk – 40 GB; Recommended 64 GB or more.
- v. Ethernet connection (LAN) or, a wireless adapter (Wi-Fi)

D.) software requirement

- i. A database like DBMS to store the list of authors and the articles. ii. A web browser like Chrome, Mozilla Firefox etc.
- iii. Operating System – Windows, Linux, macOS 32 bit and 64 bit
- iv. Xampp 5.4 or more

E.) Feasibility

1. Technical Feasibility: This is concerned with specifying equipment and software that will successfully satisfy the user requirement; the technical needs of the system may vary considerably, but might include: The facility to produce outputs in a given time: 1. Response time under conditions.

2. Ability to process a certain volume of transaction at a particular speed.

3. Facility to communicate data to distant location.

2. **Operational Feasibility:** It is mainly related to human organization and political aspects. The points to be considered are:

A). What changes will be brought with the system?

B). What organizational structures are distributed?

C). What new skills will be required? Do the existing staff members have these skills? If not, can they be trained due course of time

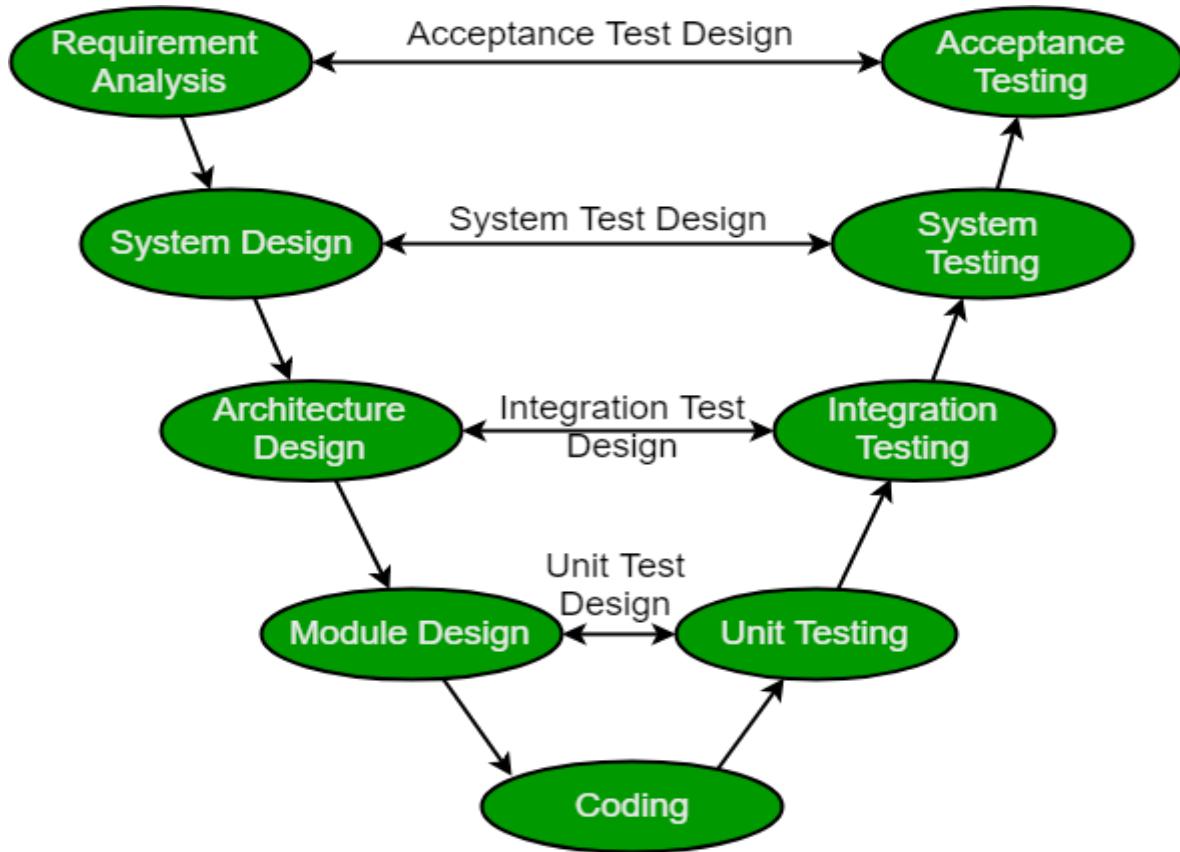
3 **Economic Feasibility:** Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More frequently known as cost/benefit system and compare them with costs. If benefits outweigh costs, a decision is taken to design and implement the system.

4. **Management Feasibility:** It is a determination of whether a proposed project will be acceptable to management. If it does not accept a project, it gives a negligible support to it; the analyst will tend to view the project as a non-feasible one.

5. **Social Feasibility:** Social feasibility is a determination of whether the project will be acceptable to the people or not. This determination typically examines the probability of the project accepted by the group directly affected by the proposed system change.

F.) Software model

The V-model is a type of SDLC model where processes execute in a sequential manner in V-shape. It is also known as Verification and Validation model. It is based on the association of a testing phase for each corresponding development stage. Development of each step directly associated with the testing phase. The next phase starts only after completion of the previous phase i.e. for each development activity, there is a testing activity corresponding to it.



Verification: It involves static analysis technique (review) done without executing code. It is the process of evaluation of the product development phase to find whether specified requirements meet.

Validation: It involves dynamic analysis technique (functional, non-functional), testing done by executing code. Validation is the process to evaluate the software after the completion of the development phase to determine whether software meets the customer expectations and requirements.

So V-Model contains Verification phases on one side of the Validation phases on the other side. Verification and Validation phases are joined by coding phase in V-shape. Thus it is called VModel.

G.) Software development life cycle

There are many models available for software development processes and many of these have been used for many years. Although, most of these models have found their places in software development processes but only few are suitable for window application program. This thesis will view the following models:

1. Waterfall model

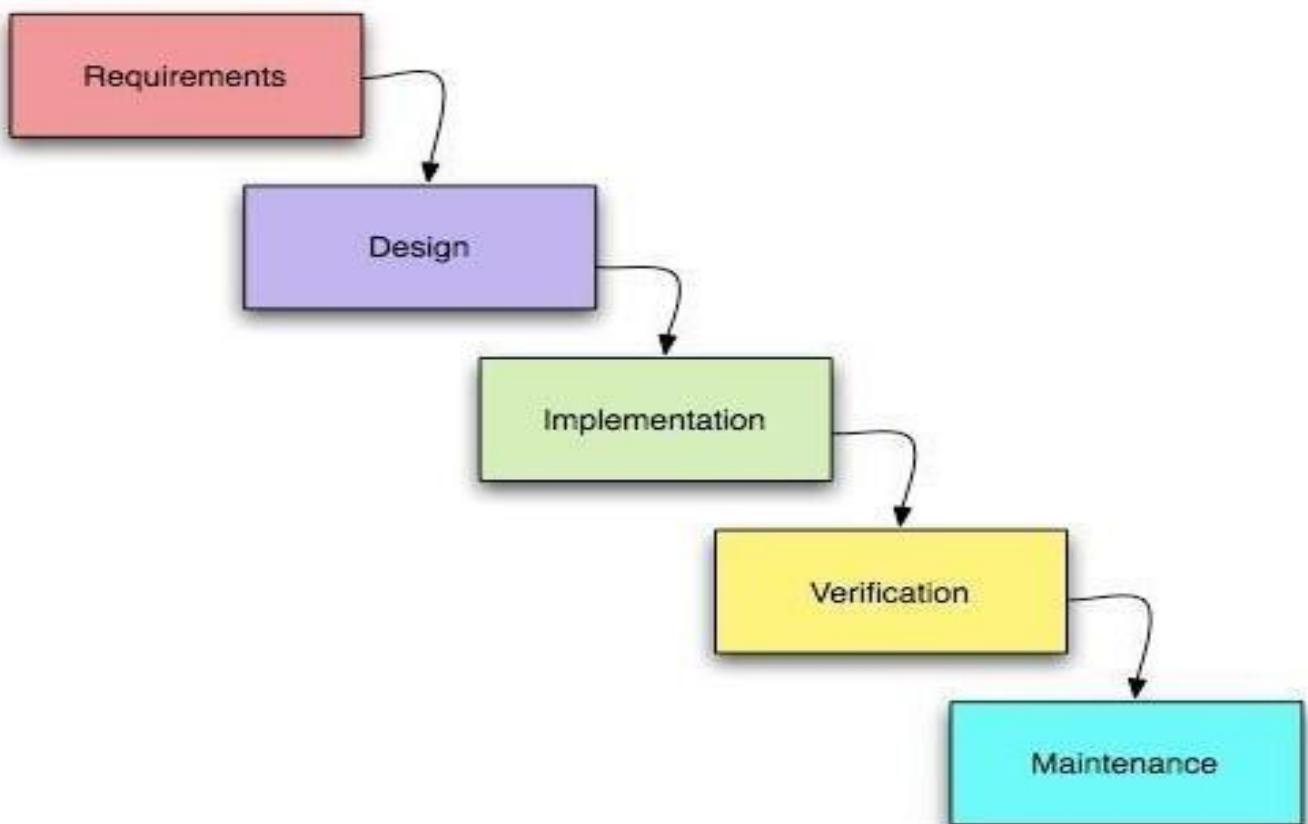
2. Iterative and incremental model

3. Spiral model

4. V-shape Model

Waterfall Model This is one of the most classical and oldest models of software engineering. It is one of the oldest models which are mostly used by many organizations for their projects in order to reach their desired goal. (Royce, W, 1970. IEEE WESCON).

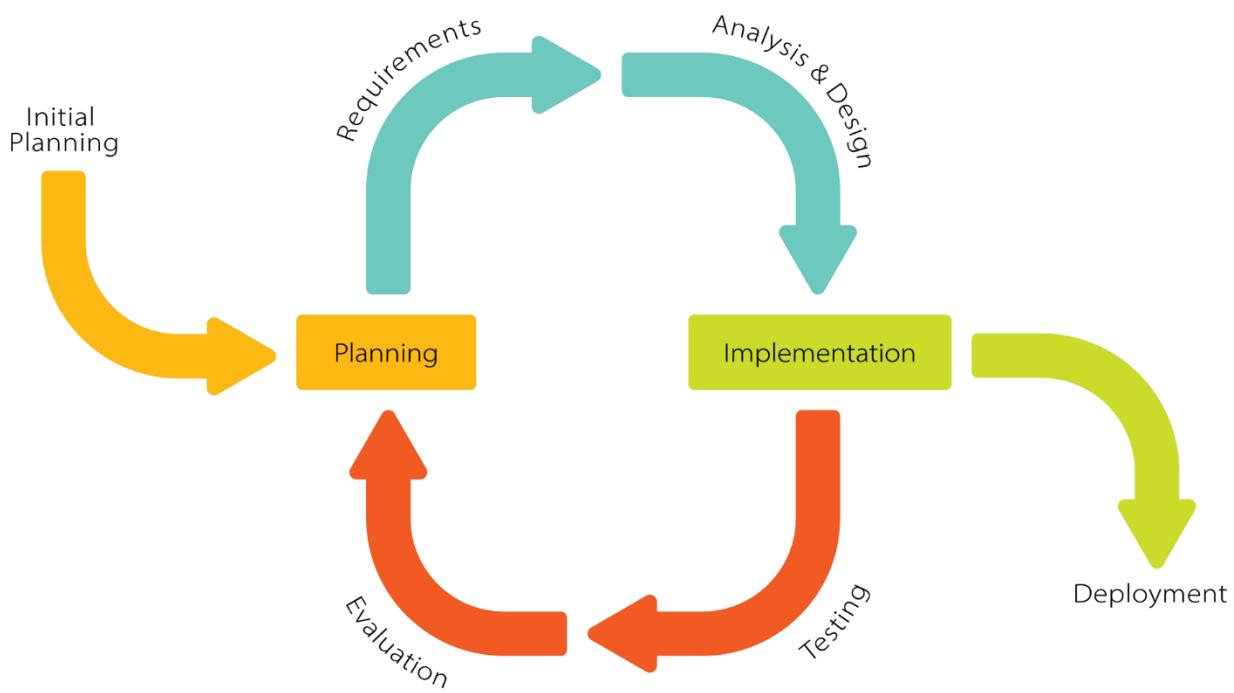
In this model, planning every phase at the early stage is important because it ensures the flow of design before they are developed. Additionally, the extensive documentation and planning of the project makes this work extremely well for projects where quality control is important. This model starts with the definition /analysis phase which is the research and brainstorming the requirements like for the project, like the software and hardware need for the project. The basic design phase comes next which is the making formulation for the required software on paper. When these two phases are agreed upon, then technical design / detail design are planned. In this part, the technical details are elaborated, functions like modules and program of each software parts are agreed upon. Thereafter, implementation phase which include coding and debugging are started before the testing phase which includes testing the whole system to make sure all the functionalities are well implemented. When it is certain that all these phases have worked out as required, the integration phase, which is putting into use the whole of the system by the company which requested the development of the software. The maintenance phase is required to ensure that this software continues to work properly as needed. Many people who are in support of this model believe that, to fix a problem at early stages of a project is more cost effective and requires less effort than doing so after months or years when the project has been completed. The figure below shows the progress flow of waterfall model (Royce, W, 1970. IEEE WESCON)



Waterfall development model (Ustudy student portal, 2012) The waterfall models do have its shortcomings because once a step is completed; one cannot go back to it to make adjustment again. If for instance after the requirement and the design phases are completed already but somehow new features are to be included in the final solution of the project, then there will be problems as long as the waterfall model is followed to the letter. But as stated in the background of this thesis, only the requirement and design of the software will be the focus point.

Iterative and incremental Model This model is developed in response to the shortcomings of waterfall model. It does not start with full specification requirement of a project, rather specify and implement some part of the software one at a time in order to review it at every step along the line to identify any further requirements. These processes have to be done again and again to produce a new requirement & version for the software and as shown in the figure below, the process starts with the initial planning, and then moves further to the real planning stage before the requirement stage. (Larman, C. & Basili, V (Dr). History of Iterative and Incremental Model 1980) Figure 2. Iterative and incremental software model (Harpal, S., 2012) Iterative development divides the deliverables into

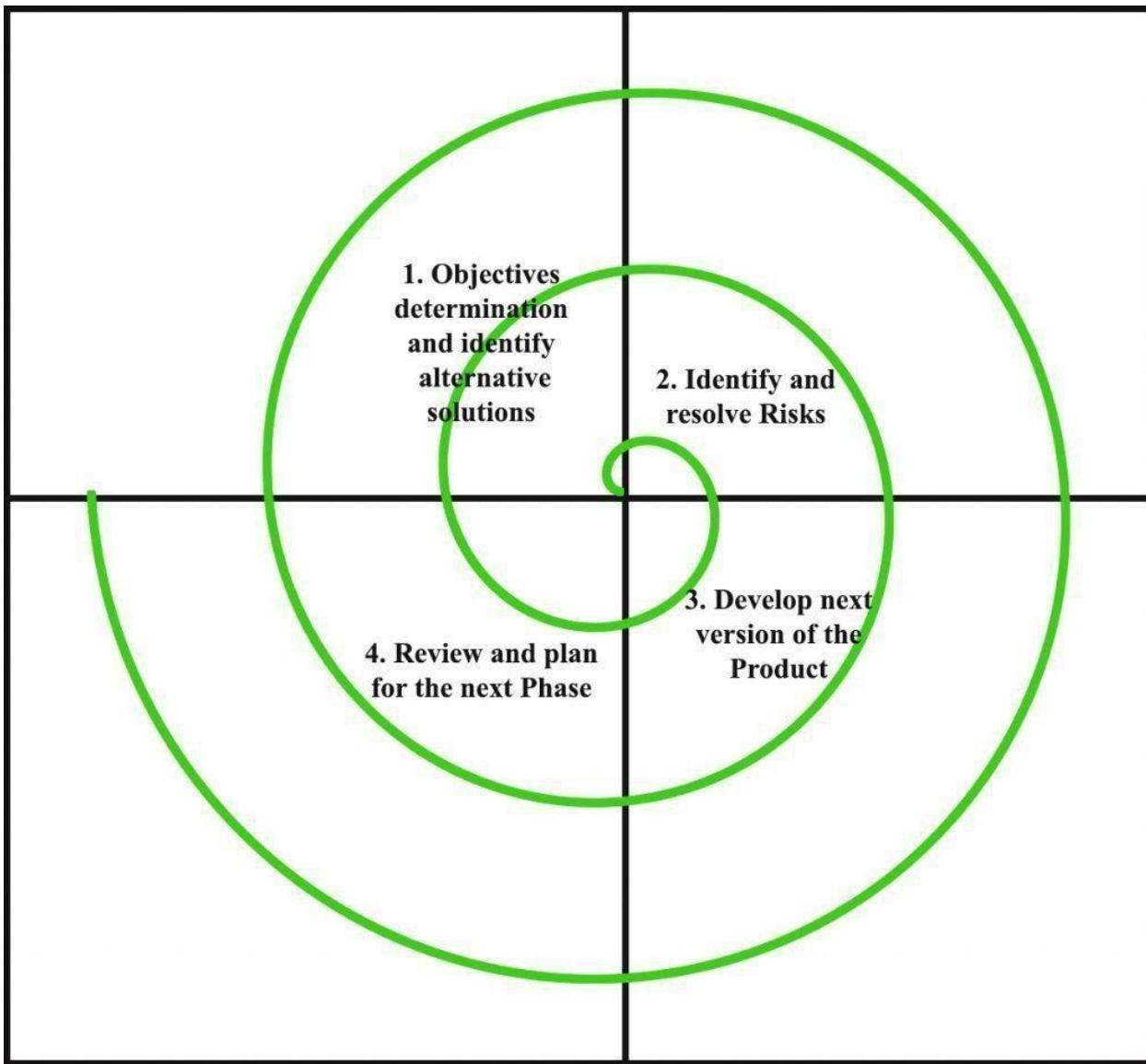
different stages. Within each stage, functionality is implemented using the cross-discipline work, which actually starts with the identification of the model to execution. (Larman, C. & Basili, V (Dr). History of Iterative and Incremental Model 1980).



Iterations models follows the phases listed below: Inception: In this phase, the scope of the project, requirements for functional and nonfunctional parts included. Elaboration: Complete detailed assessment of the project which includes risk assessment Construction: This phase is the most important and delicate part of the project because at this stage, the architectural part with the already written code from the analysis, designs, implementation and testing are incrementally inscribed into the project. Transition: The last phase of the project is the point where the system is facilitated with the operation environment Each of the aforementioned phases can be illustrated in single or multi-face iterations. These iterations are mainly time-bound instead focused on features. The role of the experts is clearly allocated and distributed, that is architects and analysts assess and write out the iteration leaving the workproduct backlog to developers and testers. (Larman, C. & Basili, V (Dr). History of Iterative and Incremental Model 1980)

Spiral Model In 1988, Barry Boehm was the first person to define the spiral model when he wrote an article, model of software development and enhancement. He defined the spiral model as a system development method (SDM) which combines the features of the prototyping model and waterfall model but it is only appropriate for some certain types of software development projects

and most often it is intended for large, expensive and complicated projects. (Boehm, B. Software Risk Management, IEEE Computer Society Press, 1989)



10 Figure 2. A spiral development model (Society for Risk Analysis (SRA), 2002)

There are series of four steps phase involved in the spiral model as shown in the figure above and they can be summarized as follows: 1. The first step of the first phase is to determine the objectives, alternatives, constraints, milestones and schedules. 2. The goal- the requirement for the software and all the constraint that might be encountered and possible alternatives that may be used during the development. 3. The risk analysis task is also assessed from the point of view of technical and management's aspect. 4. The engineering task- one or more prototype or samples of the application

is designed. The main difference between spiral model and all other software models is that the spiral model explicitly evaluates the risks involved in the development of the software. Other models do have risk management as part of their process but they do not have representation in their business process. In other model, the risk management is not part of the main tasks but just part of the sub-task. Another difference that the spiral model has is that it has no particular phase for specification requirement, design or testing although, requirement could be found and defined most likely by prototyping. Spiral model has much more advantage over other models because the developmental approach it reflects in many industries are lacking in other process model available and also uses an approach whereby some part of hardware sample phases are maintain in cases where the product being produced in a particular environment is not just software alone but also involve hardware development. Therefore, in such scenario, the developers and customer will better understand the risks involved much early and measures could be taken to correct them during the developmental stage of the soft-ware. This in particular gives spiral model more advantage over other developmental models. Some other advantages of the spiral model are listed below. (Kan, S. Metro and Model in Software Quality Engineering, 2002) Advantages 1. It is good for large and mission-critical projects 2. Most part of the software is produced early in the software life cycle 3. It has high amount of risk analysis. Spiral model is not without its own disadvantage as listed below. Disadvantages 1. It has risk assessment as an essential part of it structure even when it may not be necessary in when re-engineering or updating an already developed program. 2. Highly specific expertise is required to assess the risk analysis but it may be difficult to find an expert for such assignment. 3. This model can be very costly. 4. The success of any project that this model is used for is dependent on the risk analysis phase. 5. It does not work well for smaller projects.

12 2.4 The V-Model

This model puts a lot of emphasis on the sequential path of execution of the process. That is, each phase developed must be finished before the next one is embarked. This is exactly like the waterfall model, although this model emphasizes more on testing than the waterfall. The procedures for testing the software is developed well ahead before the codes are written for each of the phases that precede the software implementation. (Ratcliffe, A. 2011) The V-shape model starts with the requirements analysis exactly like waterfall model. The first step taken before the development phase is to create a test plan for the system which focuses a lot on meeting the functionality specified in the requirements analysis phase. The architecture and design phases along with integration test plan are all created and developed at the high-level design phase. This phase is important because the software system needs to be tested so as to know if the pieces developed so far are able to work together as a unit. However, the software components are developed at the low-level design phase alone with the unit testing required to test the functionalities of the phase. The coding of the software takes place at the implementation phase and once complete, the remaining developmental phase continues along the right hand side of V of which the testing plans which were earlier created are now put into use. The V-shape model diagram is shown in figure 3 below. It should be noted that V-shaped model also have its own advantages and disadvantages. They are listed below

This model is not flexible Figure 3. V-shape Model (THahmann Tutorial SW Development Process, 2009).

It should also be noted that V-shaped model also have its own advantages and disadvantages also which are listed below.

Advantages

1. Very simple and easy to use
2. There are specific deliverables for each of the phases
3. It has higher chances of success because of the early test plan development during the life cycle which the waterfall model does not have.
4. This is an ideal model for small projects which have requirements that can easily be understood.

Disadvantages

- 1 Developing the software during implementation phase without having to develop a prototype is a big disadvantage of this model.

2. It has a very rigid procedure just like the waterfall model.
3. ; therefore making changes can be very difficult and expensive.
4. If a problem is found during the developmental state of this model, there is no clear path provided during the testing phase.

H.) Cost estimation

The answer to how to create a hospital management system and what it would cost is dependent on a range of factors –

- Features used to build hospital information management system
- Technology stack
- Software integrations
- Development team size
- Development model – in-house vs outsourcing and location

While we have already looked into the features and technology stack, we won't be touching upon software integrations in this article because the choice of adding a new-gen technology like [blockchain in healthcare](#) or [AI in healthcare](#) is largely dependent on the project requirements. Moreover, it would increase the price range of the final hospital management software cost over a wide spectrum depending on the size of the software's application.

Let us look into the other elements on the cost end.

Details of specialists

- Backend developer: 3
- Frontend developer: 2
- UI/UX designer: 1
- QA engineer: 1
- Project manager: 1

Stage-wise cost to develop hospital management system

Good cost estimation is essential for project management success. Many costs can appear over the project management life cycle, and an accurate project cost estimation method can be the difference between a successful plan and a failed one. Project cost estimating, however, is easier said than done. Projects bring risks, and risks bring unexpected costs and cost management issues.

Project cost estimation is simplified with the help of project management software like ProjectManager. Add project budgets and planned costs for specific tasks and include labor rates for your team. When you build your plan on our Gantt chart, your estimated project costs will calculate automatically. Plus, as the project unfolds, you can track your costs in real time on our automated dashboard. Try it for free today.

Dashboard for tracking project costs

ProjectManager helps you track project costs with real-time dashboards.[Learn more](#)

What Is Project Cost Estimation?

Project cost estimation is the process that takes direct costs, indirect costs and other types of project costs into account and calculates a budget that meets the financial commitment necessary for a successful project. To do this, project managers and project estimators use a cost breakdown structure to determine all the costs in a project.

Project cost estimation is critical for any type of project, from building a bridge to developing that new killer app. Everything costs money, so the clearer you are on the amount required, the more likely you and your project team will achieve your objective.

Project cost estimating is a critical step during the project planning phase because it helps project managers create a project estimate that turns into a project budget that covers the project costs that are needed to achieve the goals and objectives of the project set forth by executives and project stakeholders.

Related: [Free Cost Estimate Template for Excel](#)

What Is a Project Estimate?

A project estimate is the process of accurately forecasting the time, cost and resources required for a project. This is done by looking at historical data, getting information from the client and itemizing each resource and its duration of use in the project.

To create a project estimate, you should first define your project scope and then create a project cost breakdown structure, which allows you to pinpoint all of your different project costs for each stage of the project life cycle.

What Is a Project Cost Breakdown Structure?

A cost breakdown structure (CBS) is a very important project costing tool that details the individual costs of a project on a document. Similar to a work breakdown structure (WBS), it's a hierarchical chart where each row represents a type of cost or item. This is done at the task level, which is called a bottom-up analysis.

Creating a cost breakdown structure might be time-consuming, but one that's worth the effort in that the result is a more accurate estimate of costs than you'd get with a top-down approach, such as basing all your estimates on the costs of previous, similar projects.

Using a cost breakdown structure is an essential part of project cost management and resource management. By zeroing in on costs at the task level early during the project planning phase, you're less likely to miss hidden costs that could come up later during the project execution stage and throw your project budget off.

Types of Project Costs

There are five main types of costs that make up your total project cost. Here's a quick overview of these types of project costs and how to measure them.

Direct costs: Direct costs are those that occur in a project and are attached to specific activities. These are generally costs that are easier to accurately estimate. They include raw materials, labor, supplies, etc.

Indirect costs: Indirect costs in a project are those that are in support of the project, such as administrative fees. These can include everything from rent to salaries of the administrative staff to utilities, etc.

Fixed costs: Fixed costs, as the name suggests, are those that don't change throughout the life cycle of a project. Some examples of fixed costs include setup costs, rental costs, insurance premiums, property taxes, etc.

Variable costs: Variable costs are costs that change due to the amount of work that's done in the project and are variable in nature. These costs can include hourly labor wages, materials, fuel costs and so on.

Sunk costs: In project cost estimating, when an investment has already been incurred and can't be recovered it's called a sunk cost or retrospective cost. Some examples of sunk costs include marketing, research, installation of new software, etc

3. Design

Database Design Database design is the process of producing a detailed data model of database. This data model contains all the need logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity. The term database design can be used to describe many different part of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structure used to store the data. In the relational model these are the tables and views. In an object database the entities and relationships map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structure, but also the forms and queries used as part of the overall database application within the database management system.

4.2 E-R Diagram of Hospital Management System An entity-relationship diagram (ERD) is an abstract and conceptual representation of data. Entityrelationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion.

Fig. 4.1: E-R Diagram of Online Marketplace Hospital Patients Patient information Admit Room Hospital Doctor record Doctor Patient Hospital On Line Appointment

25 **4.3 Database schema of Hospital Management System** A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data. A database schema can be divided broadly into two categories –

Physical Database Schema: This schema pertains to the actual storage of data and its form of storage like files, indices, etc. It defines how the data will be stored in a secondary storage.

Logical Database Schema: This schema defines all the logical constraints that need to be applied on the data stored. It defines tables, views, and integrity constraints.

List of table:

1. admin
2. Users
3. Patients
4. user_details
5. Appointment
6. Doctors
7. Doctor specialization.

A.) Input requirement

HMS helps in several functional requirements of a hospital, a few of which are as follows.

- Undertaking various registration. HMS is able to facilitate various registration needs in a hospital. ...
- Check out. ...
- Generating various reports. ...
- Database. ...
- Security. ...
- Performance. ...
- Maintainability. ...
- Reliability.

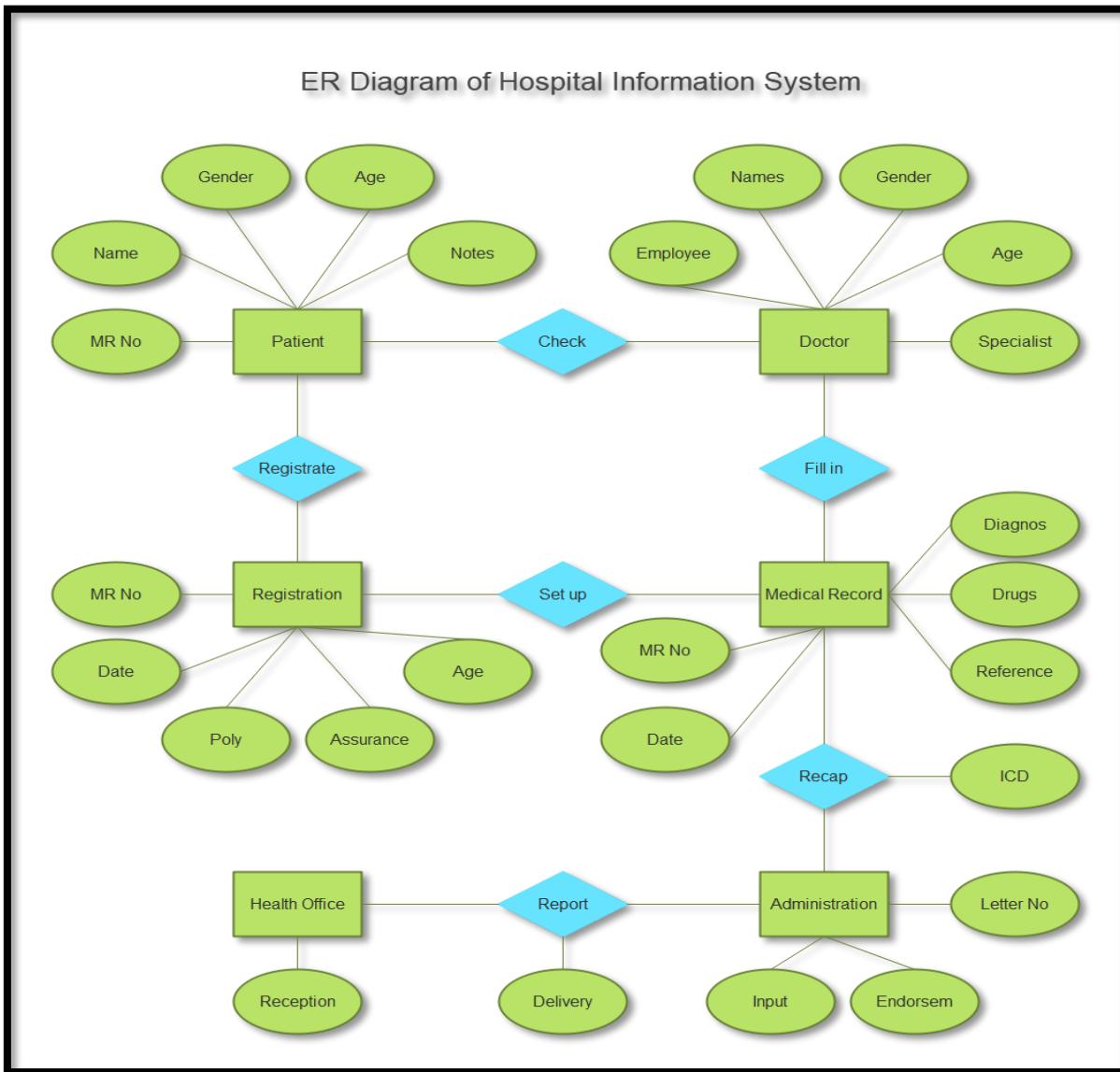
B.) Design of database

Database Design Database design is the process of producing a detailed data model of database. This data model contains all the need logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity. The term database design can be used to describe many different part of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structure used to store the data. In the relational model these are the tables and views. In an object database the entities and relationships map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structure, but also the forms and queries used as part of the overall database application within the database management system.

C.) ER- Diagram

An entity-relationship diagram (ERD) is an abstract and conceptual representation of data. Entityrelationship modeling is a database modeling method, used to produce

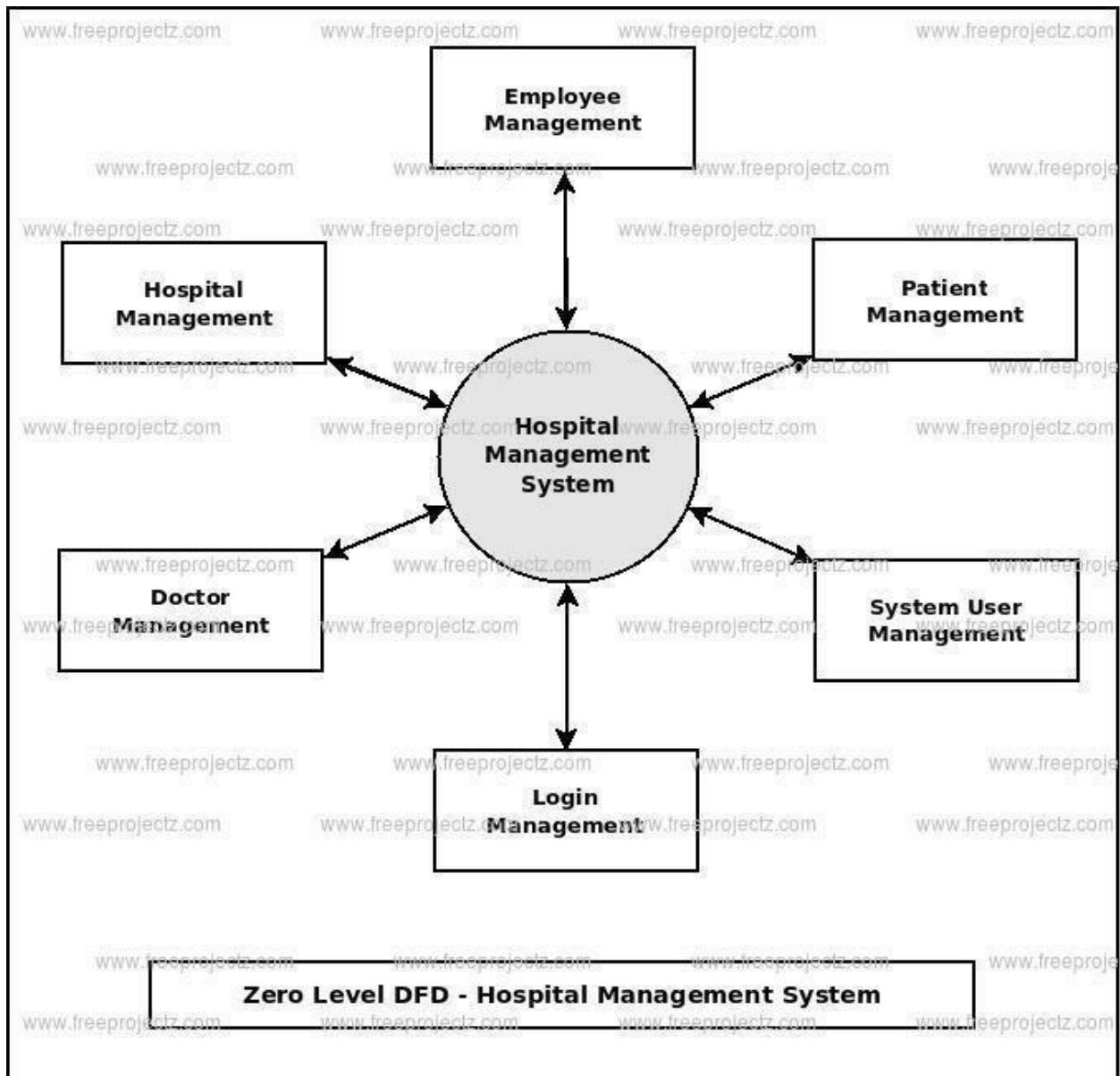
a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion.



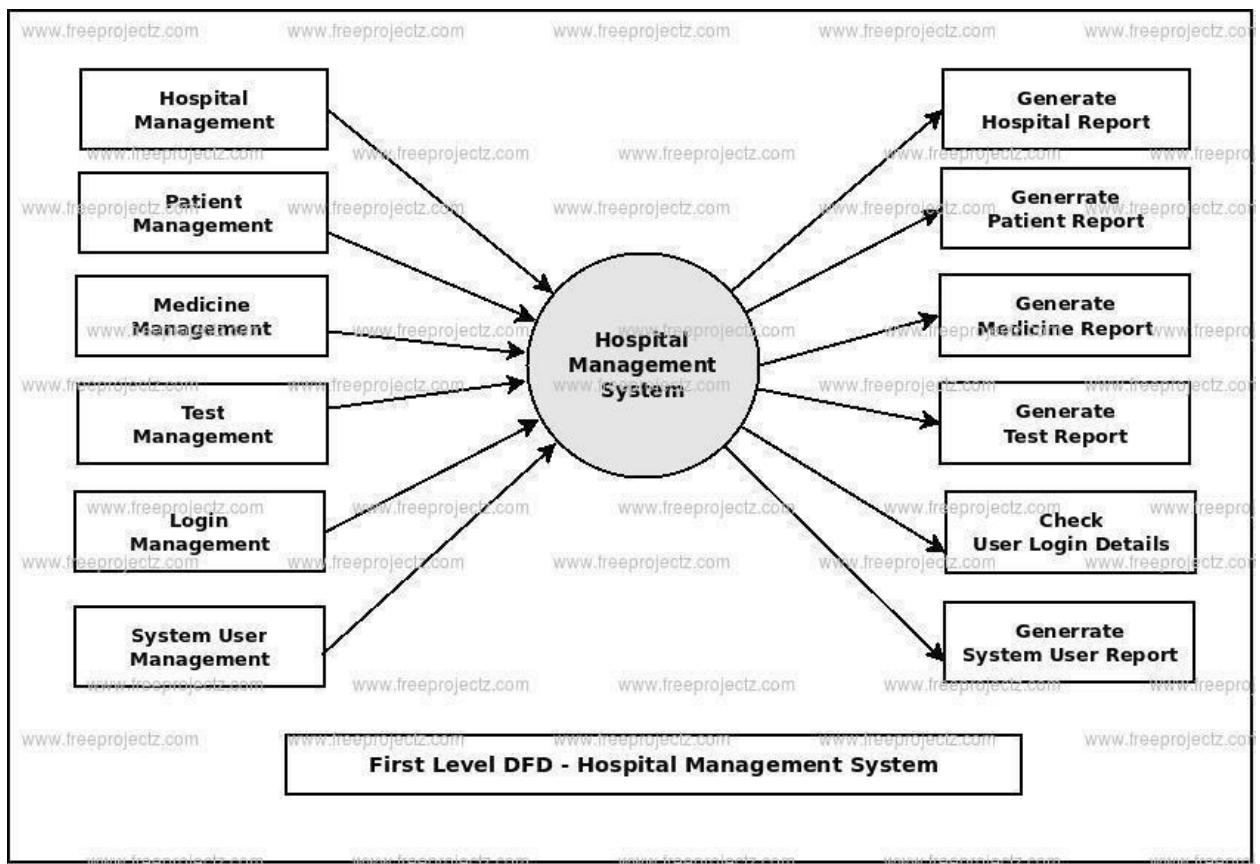
D.) Data flow diagram

The context diagram is the most abstract data flow representation of a system. It represents the entire system as a single bubble and. The various external entities with which the system interacts and the data flows occurring between the system and the external entities are also represented. The name context diagram is well justified because it represents the context in which

the system is to exist i.e. the external entities (users) that would interact with the system and specific data items they would be receiving from the system. Zero level DFD:-



First level DFD:-



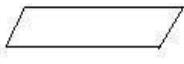
E.) System flow diagram

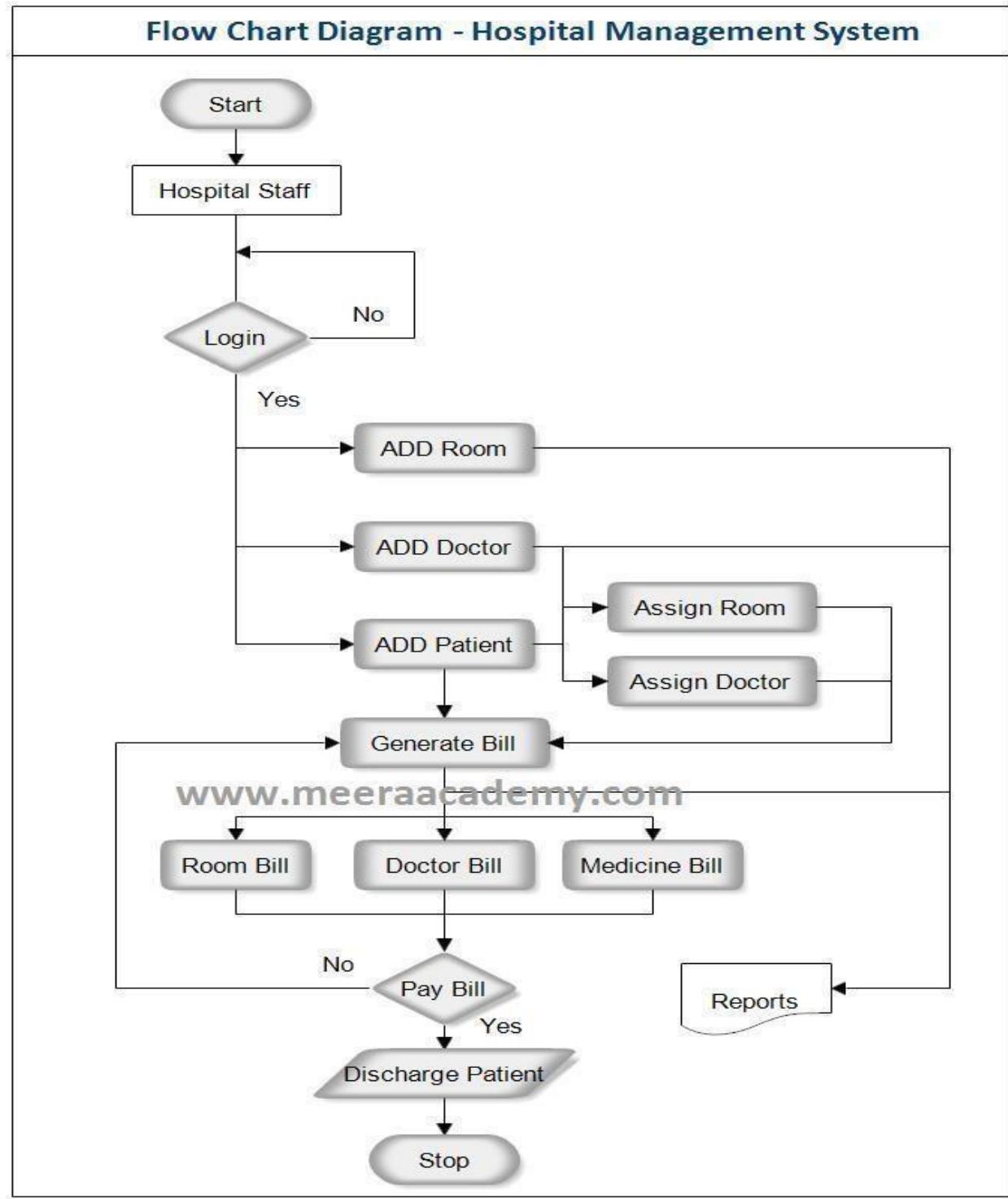
System Flow Chart for Hospital Management System

The system flow diagram is a visual representation of all processes in sequential order. The System flow chart diagram is a graphical representation of the relation between all the major parts or steps of the system. Flow chart diagram can not include minor parts of the system.

Flow Chart diagram symbols

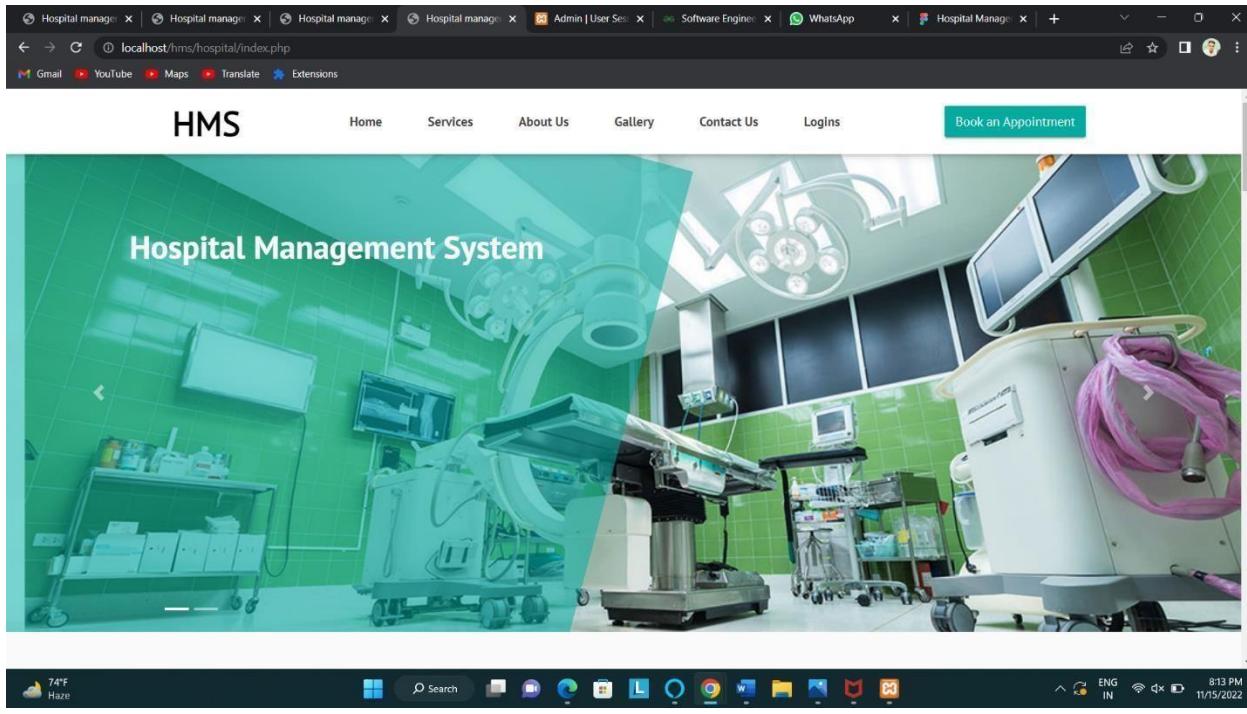
Flow Chart – Hospital Management System Project -

Symbol	Description
	Start / End : Represents the start or end point of a flowchart.
	Data Flow : Data flow are pipelines through the packets of information flow.
	Process : A Process or task performed by the system.
	Decision : Decision decides the next step will be proceed or not. Yes/No, True/False
	Input / Output : Represents input and output data.



Flow Chart for Hospital Management System

F.) user interface diagram



4. TECHNOLOGIES AND TOOLS USED

1. Visual studio code

Visual Studio Code(famously known as**VS Code**) is a free open source text editor by Microsoft. VS Code is available for Windows, Linux, and macOS. Although the editor is relatively lightweight, it includes some powerful features that have made VS Code one of the most popular development environment tools in recent times.

Visual Studio Code is a code editor in layman's terms. Visual Studio Code is “a free-editor that helps the programmer write code, helps in debugging and corrects the code using the intellisense method”. In normal terms, it facilitates users to write the code in an easy manner. Many people say that it is half of an IDE and an editor, but the decision is up to the coders. Any program/software that we see or use works on the code that runs in the background. Traditionally coding was used to do in the traditional editors or even in the basic editors like notepad! These editors used to provide basic support to the coders.

Some of them were so basic that it was very difficult in writing basic English level programs in them. As time went by, some programming languages needed a specific framework and support for further coding and development it, which was not possible using these editors. VI Editor, Sublime Text Editor, is one of the many kinds of editors that came into existence. The most prominent and which supports almost every coding language is VISUAL STUDIO CODE. Its features let the user modify the editor

as per the usage, which means the user is able to download the libraries from the internet and integrate it with the code as per his requirements.

Features

VS Code supports a wide array of programming languages from Java, C++, and Python to CSS, Go, and Dockerfile. Moreover, VS Code allows you to add on and even creating new extensions including code linters, debuggers, and cloud and web development support.

The VS Code user interface allows for a lot of interaction compared to other text editors. To simplify user experience, VS Code is divided into five main regions:

- The activity bar
 - The side bar
 - Editor groups
 - The panel
 - The status bar
- 2.php**

PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.

- PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
- PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
- It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
- PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
- PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
- PHP is forgiving: PHP language tries to be as forgiving as possible.
- PHP Syntax is C-Like.

Common uses of PHP

- PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them.
- PHP can handle forms, i.e. gather data from files, save data to a file, through email you can send data, return data to the user.
- You add, delete, modify elements within your database through PHP.
- Access cookies variables and set cookies.
- Using PHP, you can restrict users to access some pages of your website.
- It can encrypt data.

Characteristics of PHP

Five important characteristics make PHP's practical nature possible –

- Simplicity • Efficiency
- Security
- Flexibility
- Familiarity

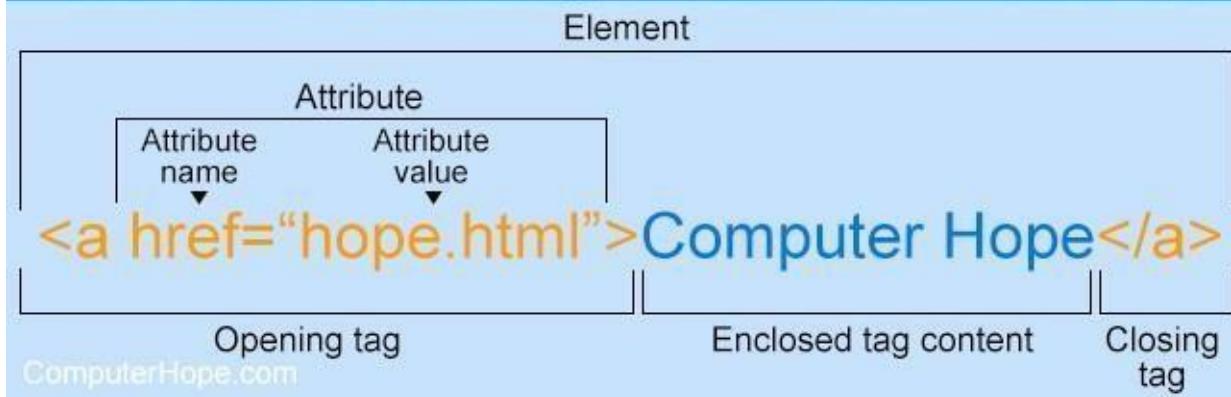
3.html

First developed by [Tim Berners-Lee](#) in [1990](#), **HTML** is short for **Hypertext Markup Language**.

HTML is used to create electronic documents (called pages) that are displayed on the [World Wide Web](#). Each page contains several connections to other pages called [hyperlinks](#). Every [web page](#) you see was written using one version of HTML.

HTML code ensures the proper formatting of text and images for your [Internet browser](#). Without HTML, a browser would not know how to display text as [elements](#) or load images or other elements. HTML also provides a basic structure of the page, upon which [Cascading Style Sheets](#) are overlaid to change its appearance. One could think of HTML as the bones (structure) of a web page, and CSS as its skin (appearance).

Technical breakdown of an HTML tag



As shown in the HTML tag example above, there aren't many components. Most HTML tags have an opening tag containing the tag name, tag [attributes](#), a closing tag containing a [forward slash](#), and the tag name being closed. For tags that do not have a closing tag like ``, it is best practice to end the tag with a forward slash.

Most tags are contained in a [less than](#) and [greater than](#) angle brackets, and everything between the open and close tag is displayed or affected by the tag. In the example above, the `<a>` tag is creating a link called "Computer Hope" that is pointing to the `hope.html` file.

4. xampp

XAMPP is an abbreviation where **X stands for Cross-Platform, A stands for Apache, M stands for MySQL, and the Ps stand for PHP and Perl**, respectively. It is an open-source package of web solutions that includes Apache distribution for many servers and command-line executables along with modules such as Apache server, [MariaDB](#), PHP, and Perl.

XAMPP helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server. It is a platform that furnishes a suitable environment to test and verify the working of projects based on Apache, Perl, MySQL database, and PHP through the system of the host itself. Among these technologies, [Perl](#) is a programming language used for web development, [PHP](#) is a backend scripting language, and MariaDB is the most vividly used database developed by MySQL. The detailed description of these components is given below.

Components of XAMPP

As defined earlier, XAMPP is used to symbolize the classification of solutions for different technologies. It provides a base for testing of projects based on different technologies through a personal server. XAMPP is an abbreviated form of each alphabet representing each of its major

components. This collection of software contains a web server named **Apache**, a database management system named **MariaDB** and scripting/ programming languages such as **PHP** and **Perl**. X denotes Cross-platform, which means that it can work on different platforms such as [Windows](#), [Linux](#), and macOS.

Many other components are also part of this collection of software and are explained below.

1. **Cross-Platform:** Different local systems have different configurations of operating systems installed in it. The component of cross-platform has been included to increase the utility and audience for this package of Apache distributions. It supports various platforms such as packages of Windows, Linus, and MAC OS.
2. **Apache:** It is an HTTP a cross-platform web server. It is used worldwide for delivering web content. The server application has made free for installation and used for the community of developers under the aegis of Apache Software Foundation. The remote server of Apache delivers the requested files, images, and other documents to the user.
3. **MariaDB:** Originally, MySQL DBMS was a part of XAMPP, but now it has been replaced by MariaDB. It is one of the most widely used relational DBMS, developed by MySQL. It offers online services of data storage, manipulation, retrieval, arrangement, and deletion.
4. **PHP:** It is the backend scripting language primarily used for web development. PHP allows users to create dynamic websites and applications. It can be installed on every platform and supports a variety of database management systems. It was implemented using C language. PHP stands for **Hypertext Processor**. It is said to be derived from Personal Home Page tools, which explains its simplicity and functionality.
5. **Perl:** It is a combination of two high-level dynamic languages, namely Perl 5 and Perl 6. Perl can be applied for finding solutions for problems based on system administration, web development, and networking. Perl allows its users to program dynamic web applications. It is very flexible and robust.
6. **phpMyAdmin:** It is a tool used for dealing with MariaDB. Its version 4.0.4 is currently being used in XAMPP. Administration of DBMS is its main role.
7. **OpenSSL:** It is the open-source implementation of the Secure Socket Layer Protocol and Transport Layer Protocol. Presently version 0.9.8 is a part of XAMPP.
8. **XAMPP Control Panel:** It is a panel that helps to operate and regulate upon other components of the XAMPP. Version 3.2.1 is the most recent update. A detailed description of the control panel will be done in the next section of the tutorial.

9. **Webalizer:** It is a Web Analytics software solution used for User logs and provide details about the usage.
10. **Mercury:** It is a mail transport system, and its latest version is 4.62. It is a mail server, which helps to manage the mails across the web.
11. **Tomcat:** Version 7.0.42 is currently being used in XAMPP. It is a servlet based on JAVA to provide JAVA functionalities.
12. **Filezilla:** It is a File Transfer Protocol Server, which supports and eases the transfer operations performed on files. Its recently updated version is 0.9.41.

XAMPP Format Support

XAMPP is supported in three file formats:

- **.EXE-** It is an extension used to denote executable files making it accessible to install because an executable file can run on a computer as any normal program.
- **.7z - 7zip file-** This extension is used to denote compressed files that support multiple data compression and encryption algorithms. It is more favored by a formalist, although it requires working with more complex files.
- **.ZIP-** This extension supports lossless compression of files. A Zipped file may contain multiple compressed files. The **Deflate algorithm** is mainly used for compression of files supported by this format. The .ZIP files are quite tricky to install as compared to .EXE

5. JAVASCRIPT

JavaScript (often shortened to JS) is a lightweight, interpreted, object-oriented language with first-class functions, and is best known as the scripting language for Web pages, but it's used in many non-browser environments as well. It is a prototype-based, multi-paradigm scripting language that is dynamic, and supports object-oriented, imperative, and functional programming styles.

JavaScript runs on the client side of the web, which can be used to design / program how the web pages behave on the occurrence of an event. JavaScript is an easy to learn and also powerful scripting language, widely used for controlling web page behavior.

Contrary to popular misconception, JavaScript is *not* "Interpreted Java". In a nutshell, JavaScript is a dynamic scripting language supporting prototype based object construction. The basic syntax is intentionally similar to both Java and C++ to reduce the number of new concepts required to learn the language. Language constructs, such

as if statements, for and while loops, and switch and try ... catch blocks function the same as in these languages (or nearly so). created programmatically in JavaScript, by attaching methods and properties to otherwise empty objects at run time, as opposed to the syntactic class definitions common in compiled languages JavaScript can function as both a [procedural](#) and an [object oriented language](#). Objects are like C++ and Java. Once an object has been constructed it can be used as a blueprint (or prototype) for creating similar objects.

JavaScript's dynamic capabilities include runtime object construction, variable parameter lists, function variables, dynamic script creation (via [eval](#)), object introspection (via for ... in), and source code recovery (JavaScript programs can decompile function bodies back into their source text).

For a more in depth discussion of JavaScript programming, follow the [JavaScript resources](#) links below.

5. CODING

The image shows two instances of Visual Studio Code running side-by-side, both displaying SQL scripts.

Top Window (hms.sql):

```
C:\>xampp>htdocs>hms>SQL File > hms.sql
1 -- phpMyAdmin SQL Dump
2 -- version 5.2.0
3 -- https://www.phpmyadmin.net/
4 --
5 -- Host: 127.0.0.1
6 -- Generation Time: Nov 08, 2022 at 03:57 AM
7 -- Server version: 10.4.24-MariaDB
8 -- PHP Version: 7.4.29
9
10 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 START TRANSACTION;
12 SET time_zone = "+00:00";
13
14
15 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */
16 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */
17 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */
18 /*!40101 SET NAMES utf8mb4 */;
19
20 --
21 -- Database: 'hms'
22 --
23
24 --
25
26 --
27 -- Table structure for table `admin`
28 --
29
30 CREATE TABLE `admin` (
31   `id` int(11) NOT NULL,
32   `username` varchar(255) NOT NULL,
33   `password` varchar(255) NOT NULL,
34   `updationDate` varchar(255) NOT NULL
35 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
36
37 --
```

Bottom Window (appointment.sql):

```
C:\>xampp>htdocs>hms>SQL File > appointment.sql
37 --
38 -- Dumping data for table `admin`
39 --
40
41 INSERT INTO `admin`(`id`, `username`, `password`, `updationDate`) VALUES
42 (1, 'admin', 'Test@12345', '30-10-2022 11:42:05 AM');
43
44 --
45
46 -- Table structure for table `appointment`
47 --
48
49
50 CREATE TABLE `appointment` (
51   `id` int(11) NOT NULL,
52   `doctorSpecialization` varchar(255) DEFAULT NULL,
53   `doctorId` int(11) DEFAULT NULL,
54   `userId` int(11) DEFAULT NULL,
55   `consultancyFees` int(11) DEFAULT NULL,
56   `appointmentDate` varchar(255) DEFAULT NULL,
57   `appointmentTime` varchar(255) DEFAULT NULL,
58   `postingDate` timestamp NULL DEFAULT current_timestamp(),
59   `userStatus` int(11) DEFAULT NULL,
60   `doctorStatus` int(11) DEFAULT NULL,
61   `updationDate` timestamp NULL DEFAULT NULL ON UPDATE current_timestamp()
62 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
63
64
65 -- Dumping data for table `appointment`
66
67
68 INSERT INTO `appointment`(`id`, `doctorSpecialization`, `doctorId`, `userId`, `consultancyFees`, `appointmentDate`, `appointmentTime`, `postingDate`, `userStatus`, `doctorStatus`, `updationDate`)
69 (1, 'ENT', 1, 1, 500, '2022-11-10', '12:45 PM', '2022-11-06 12:21:48', 1, 0, '2022-11-06 12:23:35'),
70 (2, 'ENT', 1, 2, 500, '2022-11-17', '7:00 PM', '2022-11-06 13:16:18', 1, 1, NULL);
71
72 --
```

File Edit Selection View Go Run Terminal Help hms.sql - Visual Studio Code

```

C:\xampp\htdocs\hms>SQL File > hms.sql
72 --
73 --
74 --
75 -- Table structure for table `doctors`
76 --
77
78 CREATE TABLE `doctors` (
79     `id` int(11) NOT NULL,
80     `specilization` varchar(255) DEFAULT NULL,
81     `doctorName` varchar(255) DEFAULT NULL,
82     `address` longtext DEFAULT NULL,
83     `docFees` varchar(255) DEFAULT NULL,
84     `contactno` bigint(11) DEFAULT NULL,
85     `docEmail` varchar(255) DEFAULT NULL,
86     `password` varchar(255) DEFAULT NULL,
87     `creationDate` timestamp NULL DEFAULT current_timestamp(),
88     `updateDate` timestamp NULL DEFAULT NULL ON UPDATE current_timestamp()
89 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
90 --
91 --
92 -- Dumping data for table `doctors`
93 --
94
95 INSERT INTO `doctors` (`id`, `specilization`, `doctorName`, `address`, `docFees`, `contactno`, `docEmail`, `password`, `creationDate`, `updateDate`) VALUES
96 (1, 'ENT', 'Anuj kumar', 'A 123 XYZ Apartment Raj Nagar Ext Ghaziabad', '800', 142536250, 'anujk12@test.com', 'f925916e2754e5e03f75dd5a5733251', '2022-10-18 16:52', '2022-10-18 16:52'),
97 (2, 'Endocrinologists', 'charu dua', 'X 1212 ABC Apartment Laxmi Nagar New Delhi', '800', 1231231230, 'charudu12@test.com', 'f925916e2754e5e03f75dd5a5733251', '2022-10-18 16:52', '2022-10-18 16:52')
98
99 --
100 --
101 -- Table structure for table `doctorslog`
102 --
103
104 CREATE TABLE `doctorslog` (
105     `id` int(11) NOT NULL,
106     `uid` int(11) DEFAULT NULL,
107     `username` varchar(255) DEFAULT NULL,
108
109     username varchar(255) DEFAULT NULL,
110     userip binary(16) DEFAULT NULL,
111     loginTime timestamp NULL DEFAULT current_timestamp(),
112     logout varchar(255) DEFAULT NULL,
113     status int(11) DEFAULT NULL
114 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
115 --
116 --
117 -- Dumping data for table `doctorslog`
118 --
119
120 INSERT INTO `doctorslog` (`id`, `uid`, `username`, `userip`, `loginTime`, `logout`, `status`) VALUES
121 (20, NULL, 'gfrdgf', 0x3a3a3100000000000000000000000000, '2022-11-06 01:02:16', NULL, 0),
122 (21, 2, 'charudu12@test.com', 0x3a3a3100000000000000000000000000, '2022-11-06 11:59:40', '06-11-2022 05:35:18 PM', 1),
123 (22, 2, 'charudu12@test.com', 0x3a3a3100000000000000000000000000, '2022-11-06 12:06:37', '06-11-2022 05:36:40 PM', 1),
124 (23, 2, 'charudu12@test.com', 0x3a3a3100000000000000000000000000, '2022-11-06 12:08:56', '06-11-2022 05:42:53 PM', 1),
125 (24, 1, 'anujk12@test.com', 0x3a3a3100000000000000000000000000, '2022-11-06 12:23:18', '06-11-2022 05:53:06 PM', 1),
126 (25, 2, 'charudu12@test.com', 0x3a3a3100000000000000000000000000, '2022-11-06 13:16:53', '06-11-2022 06:47:07 PM', 1),
127 (26, 1, 'anujk123@test.com', 0x3a3a3100000000000000000000000000, '2022-11-06 13:17:33', '06-11-2022 06:50:28 PM', 1);
128
129 --
130 --
131 -- Table structure for table `doctorspecilization`
132 --
133
134 CREATE TABLE `doctorspecilization` (
135     `id` int(11) NOT NULL,
136     `specilization` varchar(255) DEFAULT NULL,
137     `creationDate` timestamp NULL DEFAULT current_timestamp(),
138     `updateDate` timestamp NULL DEFAULT NULL ON UPDATE current_timestamp()
139 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
140 --
141 --
142 -- Dumping data for table `doctorspecilization`
143 --
144

```

83°F Haze

File Edit Selection View Go Run Terminal Help hms.sql - Visual Studio Code

83°F Haze

File Edit Selection View Go Run Terminal Help hms.sql - Visual Studio Code

83°F Haze

hms.sql - Visual Studio Code

```

C:\xampp\htdocs\hms>SQL File > hms.sql

144
145     INSERT INTO `doctorspecialization` (`id`, `specilization`, `creationDate`, `updationDate`) VALUES
146     (1, 'orthopedics', '2022-10-30 18:09:46', NULL),
147     (2, 'Internal Medicine', '2022-10-30 18:09:57', NULL),
148     (3, 'Obstetrics and Gynecology', '2022-10-30 18:10:18', NULL),
149     (4, 'Dermatology', '2022-10-30 18:10:28', NULL),
150     (5, 'Pediatrics', '2022-10-30 18:10:37', NULL),
151     (6, 'Radiology', '2022-10-30 18:10:46', NULL),
152     (7, 'General Surgery', '2022-10-30 18:10:56', NULL),
153     (8, 'Ophthalmology', '2022-10-30 18:11:03', NULL),
154     (9, 'Anesthesia', '2022-10-30 18:11:15', NULL),
155     (10, 'Pathology', '2022-10-30 18:11:22', NULL),
156     (11, 'EMI', '2022-10-30 18:11:30', NULL),
157     (12, 'Dental Care', '2022-10-30 18:11:39', NULL),
158     (13, 'Dermatologists', '2022-10-30 18:12:02', NULL),
159     (14, 'Endocrinologists', '2022-10-30 18:12:10', NULL),
160     (15, 'Neurologists', '2022-10-30 18:12:30', NULL);
161
162 --
163
164 --
165 -- Table structure for table `tblcontactus`
166 --
167
168 CREATE TABLE `tblcontactus` (
169     `id` int(11) NOT NULL,
170     `fullname` varchar(255) DEFAULT NULL,
171     `email` varchar(255) DEFAULT NULL,
172     `contactno` bigint(12) DEFAULT NULL,
173     `message` mediumtext DEFAULT NULL,
174     `PostingDate` timestamp NULL DEFAULT current_timestamp(),
175     `AdminRemark` mediumtext DEFAULT NULL,
176     `LastupdationDate` timestamp NULL DEFAULT NULL ON UPDATE current_timestamp(),
177     `IsRead` int(11) DEFAULT NULL
178 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
179
180 --

```

In 1, Col 1 Spaces: 2 UTF-8 LF SQL R Q

83°F Haze

File Edit Selection View Go Run Terminal Help hms.sql - Visual Studio Code

```

C:\xampp\htdocs\hms>SQL File > hms.sql

181 -- Dumping data for table `tblcontactus`
182 --
183
184 INSERT INTO `tblcontactus` (`id`, `fullname`, `email`, `contactno`, `message`, `PostingDate`, `AdminRemark`, `LastupdationDate`, `IsRead`) VALUES
185 (1, 'Anuj kumar', 'anujk30@test.com', 1425362514, 'This is for testing purposes. This is for testing purposes.This is for testing purposes.This is for testing purposes.', '2022-11-06 13:13:41', 'Contact the patient', '2022-11-06 13:13:57', 1);
186 (2, 'Anuj kumar', 'ak@gmail.com', 1111122233, 'This is for testing', '2022-11-06 13:13:41', 'Contact the patient', '2022-11-06 13:13:57', 1);
187
188 --
189
190 --
191 -- Table structure for table `tblmedicalhistory`
192 --
193
194 CREATE TABLE `tblmedicalhistory` (
195     `ID` int(10) NOT NULL,
196     `PatientID` int(10) DEFAULT NULL,
197     `BloodPressure` varchar(200) DEFAULT NULL,
198     `BloodSugar` varchar(200) NOT NULL,
199     `Weight` varchar(100) DEFAULT NULL,
200     `Temperature` varchar(200) DEFAULT NULL,
201     `MedicalPres` mediumtext DEFAULT NULL,
202     `CreationDate` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
203 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
204
205 --
206 -- Dumping data for table `tblmedicalhistory`
207 --
208
209 INSERT INTO `tblmedicalhistory` (`ID`, `PatientID`, `BloodPressure`, `BloodSugar`, `Weight`, `Temperature`, `MedicalPres`, `CreationDate`) VALUES
210 (1, 1, '80/120', '120', '95', '98', 'test', '2022-11-06 13:19:41');
211
212 --
213
214 --
215 -- Table structure for table `tblpage`
216 --
217

```

In 1, Col 1 Spaces: 2 UTF-8 LF SQL R Q

83°F Haze

File Edit Selection View Go Run Terminal Help hms.sql - Visual Studio Code

File Edit Selection View Go Run Terminal Help

hms.sql - Visual Studio Code

```

C:\xampp>htdocs>hms>SQL File > hms.sql

217 CREATE TABLE `tblpage` (
218     `ID` int(10) NOT NULL,
219     `PageType` varchar(200) DEFAULT NULL,
220     `PageTitle` varchar(200) DEFAULT NULL,
221     `PageDescription` mediumtext DEFAULT NULL,
222     `Email` varchar(120) DEFAULT NULL,
223     `MobileNumber` bigint(10) DEFAULT NULL,
224     `UpdationDate` timestamp NULL DEFAULT current_timestamp(),
225     `OpeningTime` varchar(255) DEFAULT NULL
226 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
227
228 --
229 -- Dumping data for table `tblpage`
230 --
231
232 INSERT INTO `tblpage` (`ID`, `PageType`, `PageTitle`, `PageDescription`, `Email`, `MobileNumber`, `UpdationDate`, `OpeningTime`) VALUES
233 (1, 'aboutus', 'About Us', 'padding: 0px; margin-right: 0px; margin-bottom: 1.31em; margin-left: \\"\\n roman\\"; font-size: \\"\\n',
234 (2, 'contactus', 'Contact Details', 'D-204, Hole Town South West, Delhi-110096, India', 'info@gmail.com', 1122334455, '2020-05-20 07:24:07', '9 am to 8 pm');
235
236 --
237 --
238
239 --
240 -- Table structure for table `tblpatient`
241 --
242
243 CREATE TABLE `tblpatient` (
244     `ID` int(10) NOT NULL,
245     `Docid` int(10) DEFAULT NULL,
246     `PatientName` varchar(200) DEFAULT NULL,
247     `PatientContno` bigint(10) DEFAULT NULL,
248     `PatientEmail` varchar(200) DEFAULT NULL,
249     `PatientGender` varchar(50) DEFAULT NULL,
250     `PatientAdd` mediumtext DEFAULT NULL,
251     `PatientAge` int(10) DEFAULT NULL,
252     `PatientMedhis` mediumtext DEFAULT NULL,
253     `CreationDate` timestamp NULL DEFAULT current_timestamp(),
254
255 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
256
257 --
258 -- Dumping data for table `tblpatient`
259 --
260
261 INSERT INTO `tblpatient` (`ID`, `Docid`, `PatientName`, `PatientContno`, `PatientEmail`, `PatientGender`, `PatientAdd`, `PatientAge`, `PatientMedhis`, `CreationDate`, `Up
262 (1, 1, 'Amit Kumar', 1231231230, 'amitk@gmail.com', 'male', 'New Delhi india', 35, 'NA', '2022-11-06 13:18:31', NULL);
263
264 --
265
266 --
267 -- Table structure for table `userlog`
268 --
269
270 CREATE TABLE `userlog` (
271     `id` int(11) NOT NULL,
272     `uid` int(11) DEFAULT NULL,
273     `username` varchar(255) DEFAULT NULL,
274     `userip` binary(16) DEFAULT NULL,
275     `loginime` timestamp NULL DEFAULT current_timestamp(),
276     `logout` varchar(255) DEFAULT NULL,
277     `status` int(11) DEFAULT NULL
278 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
279
280 --
281 -- Dumping data for table `userlog`
282 --
283
284 INSERT INTO `userlog` (`id`, `uid`, `username`, `userip`, `loginime`, `logout`, `status`) VALUES
285 (1, 1, 'johndoe12@test.com', 0x3a3a3100000000000000000000000000, '2022-11-06 12:14:11', NULL, 1),
286 (2, 1, 'johndoe12@test.com', 0x3a3a3100000000000000000000000000, '2022-11-06 12:21:20', '06-11-2022 05:53:00 PM', 1),
287 (3, NULL, 'amitk@gmail.com', 0x3a3a3100000000000000000000000000, '2022-11-06 13:15:43', NULL, 0),
288 (4, 2, 'amitk@gmail.com', 0x3a3a3100000000000000000000000000, '2022-11-06 13:15:58', '06-11-2022 06:50:46 PM', 1);
289

```

83°F Haze

File Edit Selection View Go Run Terminal Help

hms.sql - Visual Studio Code

```

C:\xampp>htdocs>hms>SQL File > hms.sql

253     `CreationDate` timestamp NULL DEFAULT current_timestamp(),
254     `UpdationDate` timestamp NULL DEFAULT NULL ON UPDATE current_timestamp()
255 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
256
257 --
258 -- Dumping data for table `tblpatient`
259 --
260
261 INSERT INTO `tblpatient` (`ID`, `Docid`, `PatientName`, `PatientContno`, `PatientEmail`, `PatientGender`, `PatientAdd`, `PatientAge`, `PatientMedhis`, `CreationDate`, `Up
262 (1, 1, 'Amit Kumar', 1231231230, 'amitk@gmail.com', 'male', 'New Delhi india', 35, 'NA', '2022-11-06 13:18:31', NULL);
263
264 --
265
266 --
267 -- Table structure for table `userlog`
268 --
269
270 CREATE TABLE `userlog` (
271     `id` int(11) NOT NULL,
272     `uid` int(11) DEFAULT NULL,
273     `username` varchar(255) DEFAULT NULL,
274     `userip` binary(16) DEFAULT NULL,
275     `loginime` timestamp NULL DEFAULT current_timestamp(),
276     `logout` varchar(255) DEFAULT NULL,
277     `status` int(11) DEFAULT NULL
278 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
279
280 --
281 -- Dumping data for table `userlog`
282 --
283
284 INSERT INTO `userlog` (`id`, `uid`, `username`, `userip`, `loginime`, `logout`, `status`) VALUES
285 (1, 1, 'johndoe12@test.com', 0x3a3a3100000000000000000000000000, '2022-11-06 12:14:11', NULL, 1),
286 (2, 1, 'johndoe12@test.com', 0x3a3a3100000000000000000000000000, '2022-11-06 12:21:20', '06-11-2022 05:53:00 PM', 1),
287 (3, NULL, 'amitk@gmail.com', 0x3a3a3100000000000000000000000000, '2022-11-06 13:15:43', NULL, 0),
288 (4, 2, 'amitk@gmail.com', 0x3a3a3100000000000000000000000000, '2022-11-06 13:15:58', '06-11-2022 06:50:46 PM', 1);
289

```

83°F Haze

File Edit Selection View Go Run Terminal Help hms.sql - Visual Studio Code

```

hms.sql ×
C:\xampp\htdocs>sql > hms > SQL File > hms.sql
288 (4, 2, 'amit@gmail.com', 0x3a3a3100000000000000000000000000, '2022-11-06 13:15:58', '06-11-2022 06:50:46 PM', 1);
289
290 --
291
292 --
293 -- Table structure for table `users`
294 --
295 CREATE TABLE `users` (
296     `id` int(11) NOT NULL,
297     `fullName` varchar(255) DEFAULT NULL,
298     `address` longtext DEFAULT NULL,
299     `city` varchar(255) DEFAULT NULL,
300     `gender` varchar(255) DEFAULT NULL,
301     `email` varchar(255) DEFAULT NULL,
302     `password` varchar(255) DEFAULT NULL,
303     `regDate` timestamp NULL DEFAULT current_timestamp(),
304     `updationDate` timestamp NULL DEFAULT NULL ON UPDATE current_timestamp()
305 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
306
307 --
308 --
309 -- Dumping data for table `users`
310 --
311
312 INSERT INTO `users` (`id`, `fullName`, `address`, `city`, `gender`, `email`, `password`, `regDate`, `updationDate`) VALUES
313 (1, 'John Doe', 'A 123 ABC Apartment GZB 201017', 'Ghaziabad', 'male', 'johndoe12@test.com', 'f925916e2754e5e03f75dd58a5733251', '2022-11-06 12:13:56', NULL),
314 (2, 'Amit Kumar', 'new Delhi india', 'New Delhi', 'male', 'amitk@gmail.com', 'f925916e2754e5e03f75dd58a5733251', '2022-11-06 13:15:32', NULL);
315
316 --
317 -- Indexes for dumped tables
318 --
319
320 --
321 -- Indexes for table `admin`
322 --
323 ALTER TABLE `admin`
324     ADD PRIMARY KEY (`id`);

Ln 1, Col 1 Spaces: 2 UTF-8 LF SQL ⚙️
83°F Haze ENG IN 12:46 PM 4/24/2023

File Edit Selection View Go Run Terminal Help hms.sql - Visual Studio Code
hms.sql ×
C:\xampp\htdocs>sql > hms > SQL File > hms.sql
323 ALTER TABLE `admin`
324     ADD PRIMARY KEY (`id`);
325
326 --
327 -- Indexes for table `appointment`
328 --
329 ALTER TABLE `appointment`
330     ADD PRIMARY KEY (`id`);
331
332 --
333 -- Indexes for table `doctors`
334 --
335 ALTER TABLE `doctors`
336     ADD PRIMARY KEY (`id`);
337
338 --
339 -- Indexes for table `doctorslog`
340 --
341 ALTER TABLE `doctorslog`
342     ADD PRIMARY KEY (`id`);
343
344 --
345 -- Indexes for table `doctorspecialization`
346 --
347 ALTER TABLE `doctorspecialization`
348     ADD PRIMARY KEY (`id`);
349
350 --
351 -- Indexes for table `tblcontactus`
352 --
353 ALTER TABLE `tblcontactus`
354     ADD PRIMARY KEY (`id`);
355
356 --
357 -- Indexes for table `tblmedicalhistory`
358 --
359 ALTER TABLE `tblmedicalhistory`

Ln 1, Col 1 Spaces: 2 UTF-8 LF SQL ⚙️
83°F Haze ENG IN 12:46 PM 4/24/2023

```

File Edit Selection View Go Run Terminal Help

hms.sql - Visual Studio Code

```

C:\xampp>htdocs>hms>SQL File > hms.sql

359 ALTER TABLE `tblmedicalhistory`
360 | ADD PRIMARY KEY ( `ID` );
361 |
362 --
363 -- Indexes for table `tblpage`
364 --
365 ALTER TABLE `tblpage`
366 | ADD PRIMARY KEY ( `ID` );
367 |
368 --
369 -- Indexes for table `tblpatient`
370 --
371 ALTER TABLE `tblpatient`
372 | ADD PRIMARY KEY ( `ID` );
373 |
374 --
375 -- Indexes for table `userlog`
376 --
377 ALTER TABLE `userlog`
378 | ADD PRIMARY KEY ( `id` );
379 |
380 --
381 -- Indexes for table `users`
382 --
383 ALTER TABLE `users`
384 | ADD PRIMARY KEY ( `id` ),
385 | ADD KEY `email` ( `email` );
386 |
387 --
388 -- AUTO_INCREMENT for dumped tables
389 --
390 --
391 --
392 -- AUTO_INCREMENT for table `admin`
393 --
394 ALTER TABLE `admin`
395 | MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
396 |
397 --
398 -- AUTO_INCREMENT for table `appointment`
399 --
400 ALTER TABLE `appointment`
401 | MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;
402 |
403 --
404 -- AUTO_INCREMENT for table `doctors`
405 --
406 ALTER TABLE `doctors`
407 | MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;
408 |
409 --
410 -- AUTO_INCREMENT for table `doctorslog`
411 --
412 ALTER TABLE `doctorslog`
413 | MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=27;
414 |
415 --
416 -- AUTO_INCREMENT for table `doctorspecialization`
417 --
418 ALTER TABLE `doctorspecialization`
419 | MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=18;
420 |
421 --
422 -- AUTO_INCREMENT for table `tblcontactus`
423 --
424 ALTER TABLE `tblcontactus`
425 | MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;
426 |
427 --
428 -- AUTO_INCREMENT for table `tblmedicalhistory`
429 --

```

Ln 1, Col 1 Spaces: 2 UTF-8 LF SQL R

83°F Haze

File Edit Selection View Go Run Terminal Help

hms.sql - Visual Studio Code

```

393 --
394 ALTER TABLE `admin`
395 | MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
396 |
397 --
398 -- AUTO_INCREMENT for table `appointment`
399 --
400 ALTER TABLE `appointment`
401 | MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;
402 |
403 --
404 -- AUTO_INCREMENT for table `doctors`
405 --
406 ALTER TABLE `doctors`
407 | MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;
408 |
409 --
410 -- AUTO_INCREMENT for table `doctorslog`
411 --
412 ALTER TABLE `doctorslog`
413 | MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=27;
414 |
415 --
416 -- AUTO_INCREMENT for table `doctorspecialization`
417 --
418 ALTER TABLE `doctorspecialization`
419 | MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=18;
420 |
421 --
422 -- AUTO_INCREMENT for table `tblcontactus`
423 --
424 ALTER TABLE `tblcontactus`
425 | MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;
426 |
427 --
428 -- AUTO_INCREMENT for table `tblmedicalhistory`
429 --

```

Ln 1, Col 1 Spaces: 2 UTF-8 LF SQL R

83°F Haze

The screenshot shows a Visual Studio Code window with the file 'hms.sql' open. The code is an SQL script used to modify table structures, specifically setting AUTO_INCREMENT values for various tables. The script includes comments and several ALTER TABLE statements. The code is as follows:

```
C:\>xampp >htdocs >hms >SQL File > hms.sql
428 -- AUTO_INCREMENT for table `tblmedicalhistory`
429 --
430 ALTER TABLE `tblmedicalhistory`
431 | MODIFY `ID` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
432 |
433 --
434 -- AUTO_INCREMENT for table `tblpage`
435 --
436 ALTER TABLE `tblpage`
437 | MODIFY `ID` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;
438 |
439 --
440 -- AUTO_INCREMENT for table `tblpatient`
441 --
442 ALTER TABLE `tblpatient`
443 | MODIFY `ID` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
444 |
445 --
446 -- AUTO_INCREMENT for table `userlog`
447 --
448 ALTER TABLE `userlog`
449 | MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;
450 |
451 --
452 -- AUTO_INCREMENT for table `users`
453 --
454 ALTER TABLE `users`
455 | MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;
456 COMMIT;
457 
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
458 /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
459 /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

6. Implementation and result

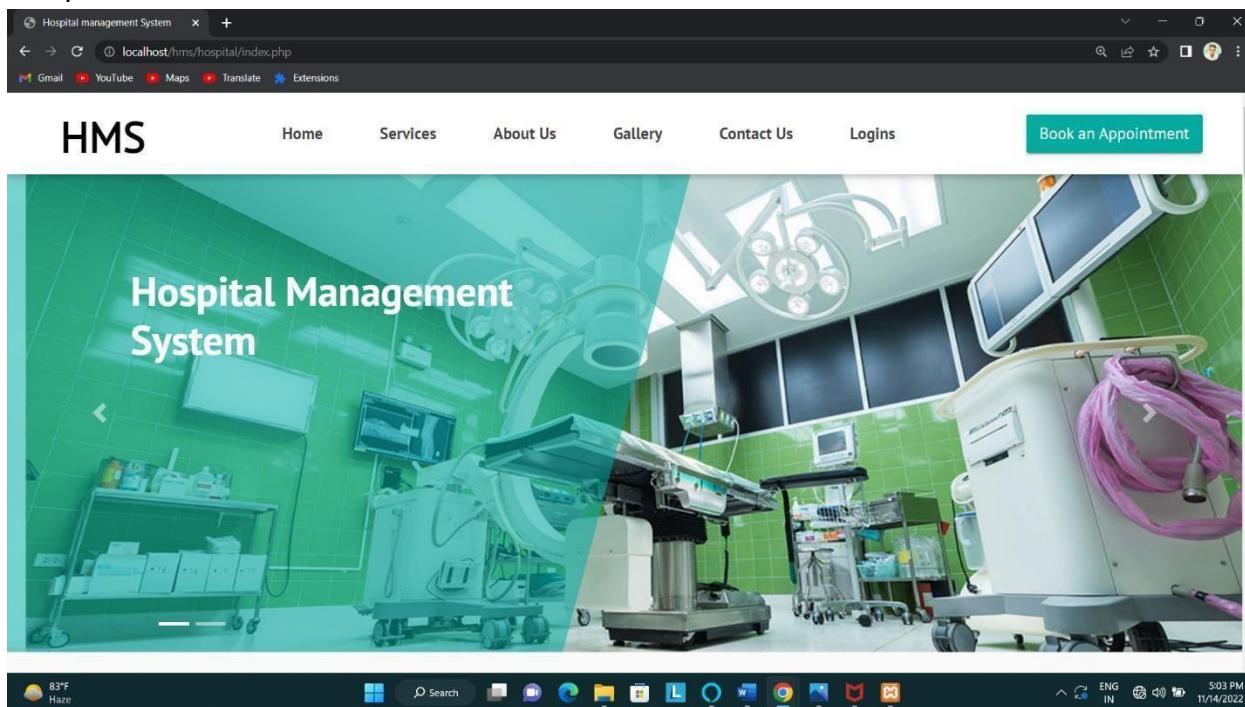
Implementation Implementation is the process of having system personal check out and provides new equipment's into use, train the user to install a new application and construct any files of data needed to use it. There are three types of implementation. Implementation of computer system to replace a manual system. To problem encountered are covering files, training user, creating accurate files and verifying print outs for integrity. Implementation of a new computer system to replace an existing one. This is usually difficult conversion. If not properly planned, there can be many problems. So large computer system many take as long as a year to convert. Implementation of a modified application to replace the existing one using the same computer. This type of conversing is relatively easy to handle, usually there are no major change in the file. Our project is yet to be implemented.

Implementation Environment :-

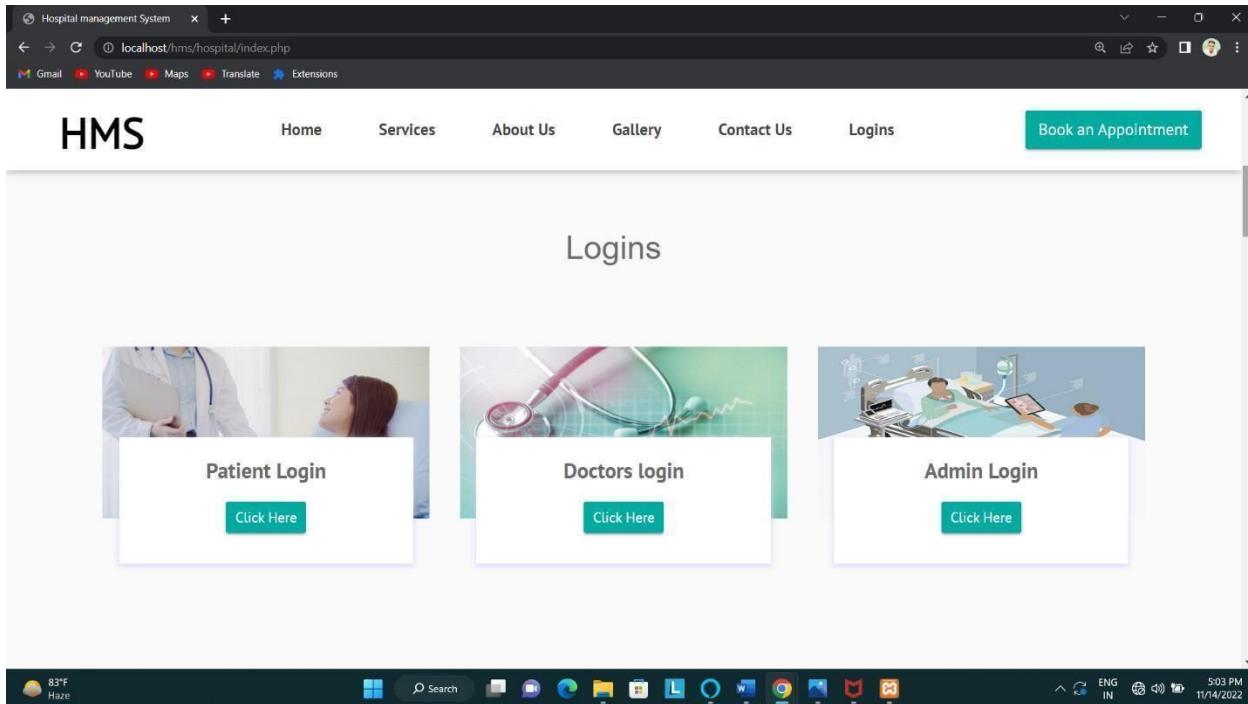
The implementation view of software requirement presents the real world manifestation of processing functions and information structures. This computerized system is specified in a manner that dictates accommodation of certain implementation details. The implementation environment of the developed system facilitates multiple users to use this system simultaneously. The user interfaces are designed keeping in mind that the users of this system are familiar to using GUI-based systems. Thus, we restricted ourselves to developing a GUI-based system so that it becomes easier for the end user to get acquainted to the developed system

-Home Page

Actor: Any users



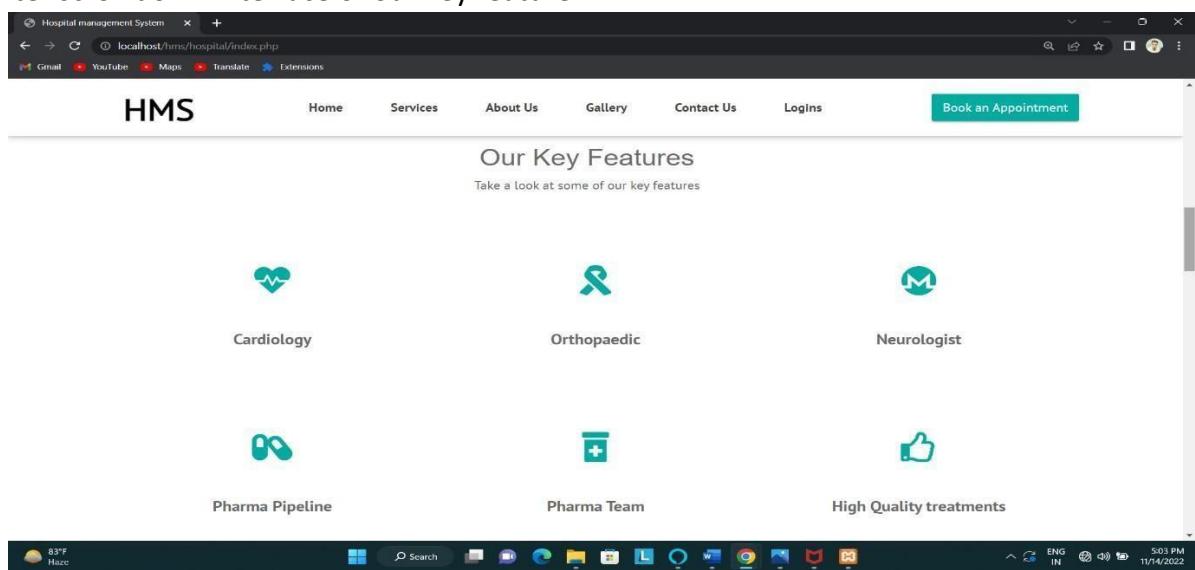
Then scroll down to login page.



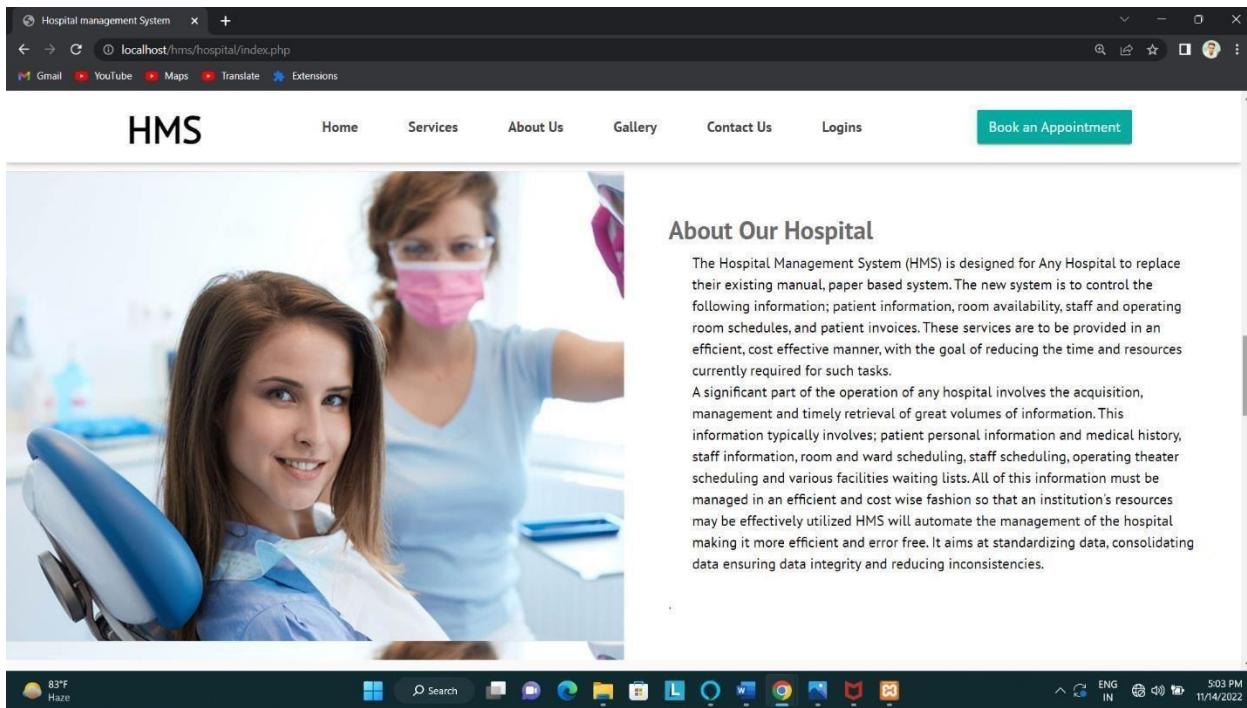
There are three option:-

1. Patient login
2. Doctors login 3. Admin login

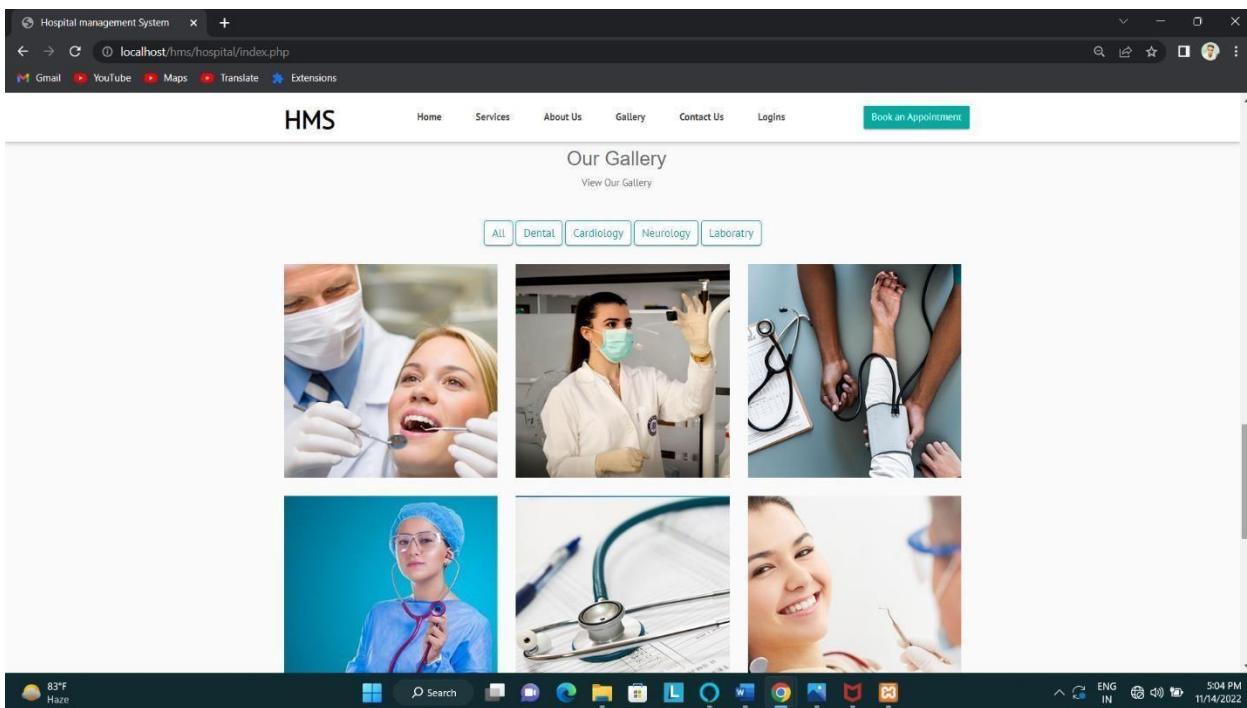
After scroll down interface of our key feature.



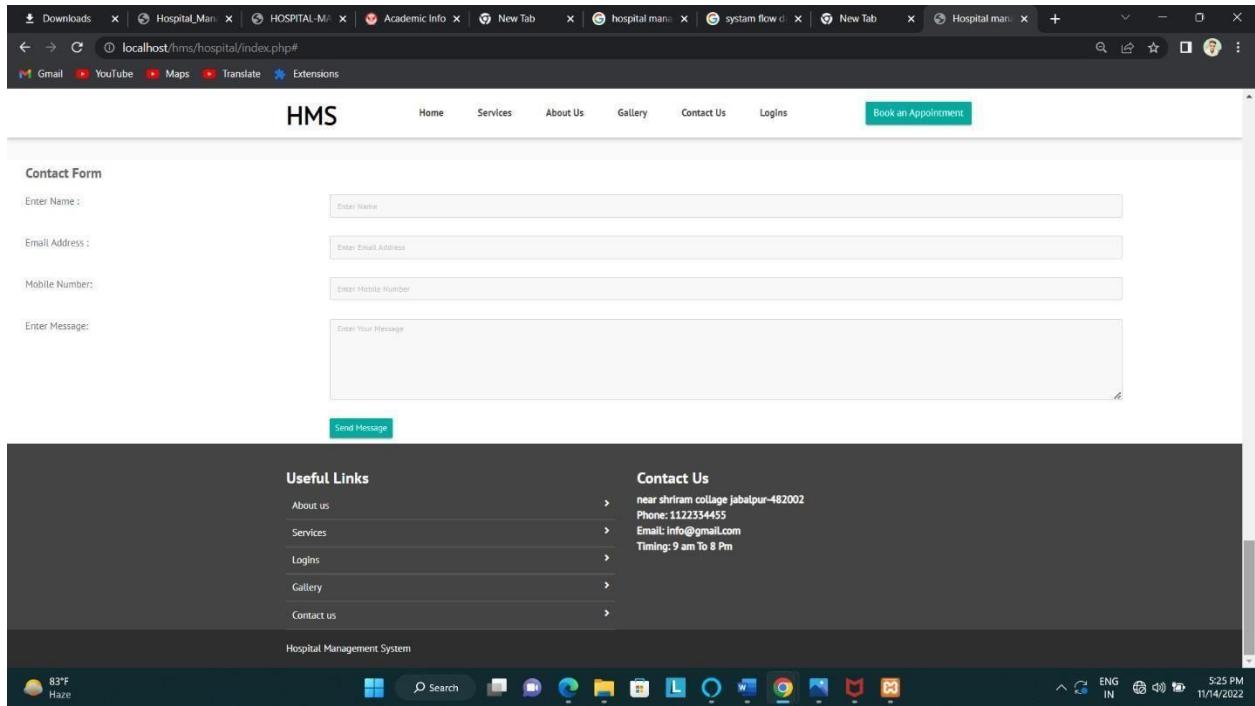
After “key feature” the “about our hospital” is opend.



After this the “our gallery” will be open. In gallery specialist are given.

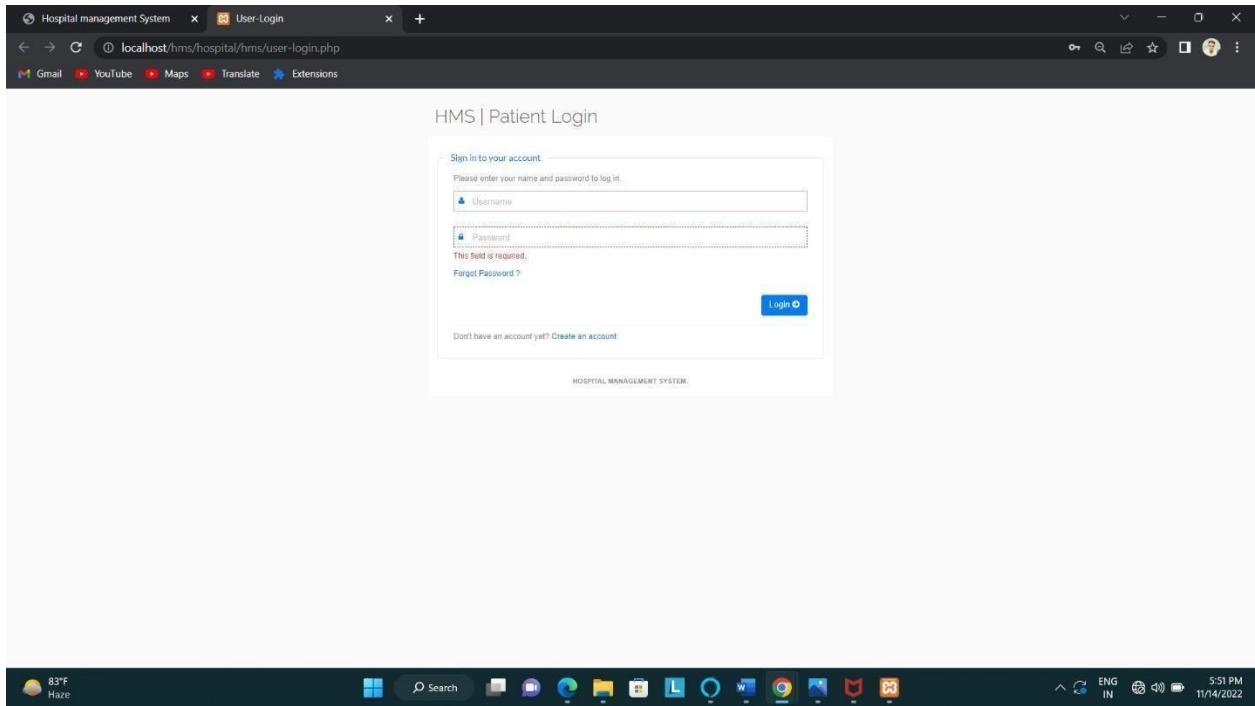


At the last “contact us” will open.



Patient login page

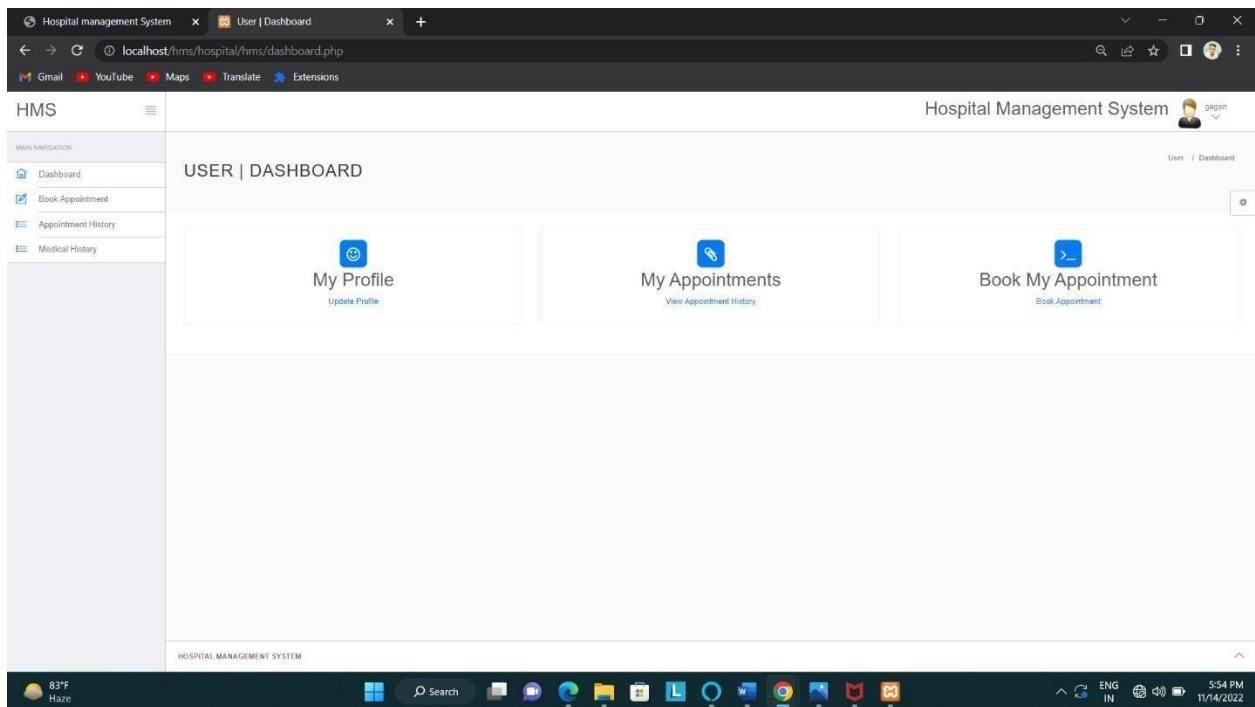
After click on patient login page.



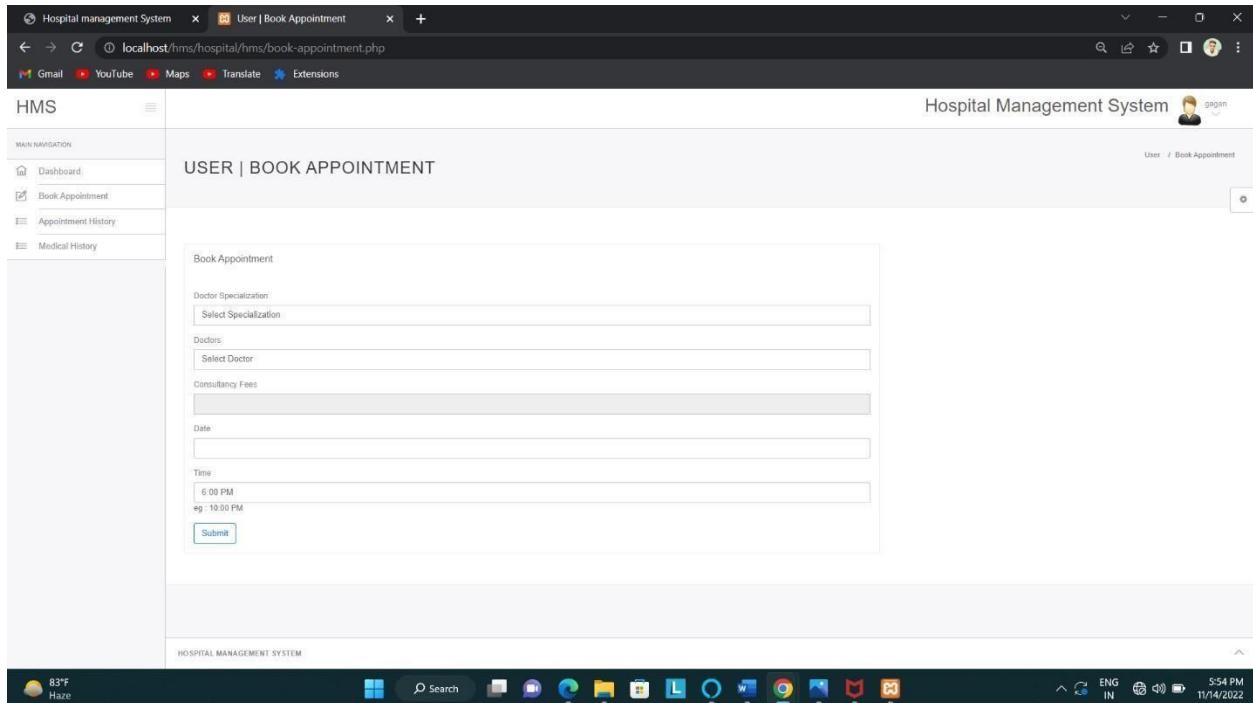
Actor: User.

Input: User email and Password

Output: User profile page



In patient dashboard you can book appointment and check appointment history and medical history.



The image displays two screenshots of a Hospital Management System (HMS) interface, likely from a Windows operating system.

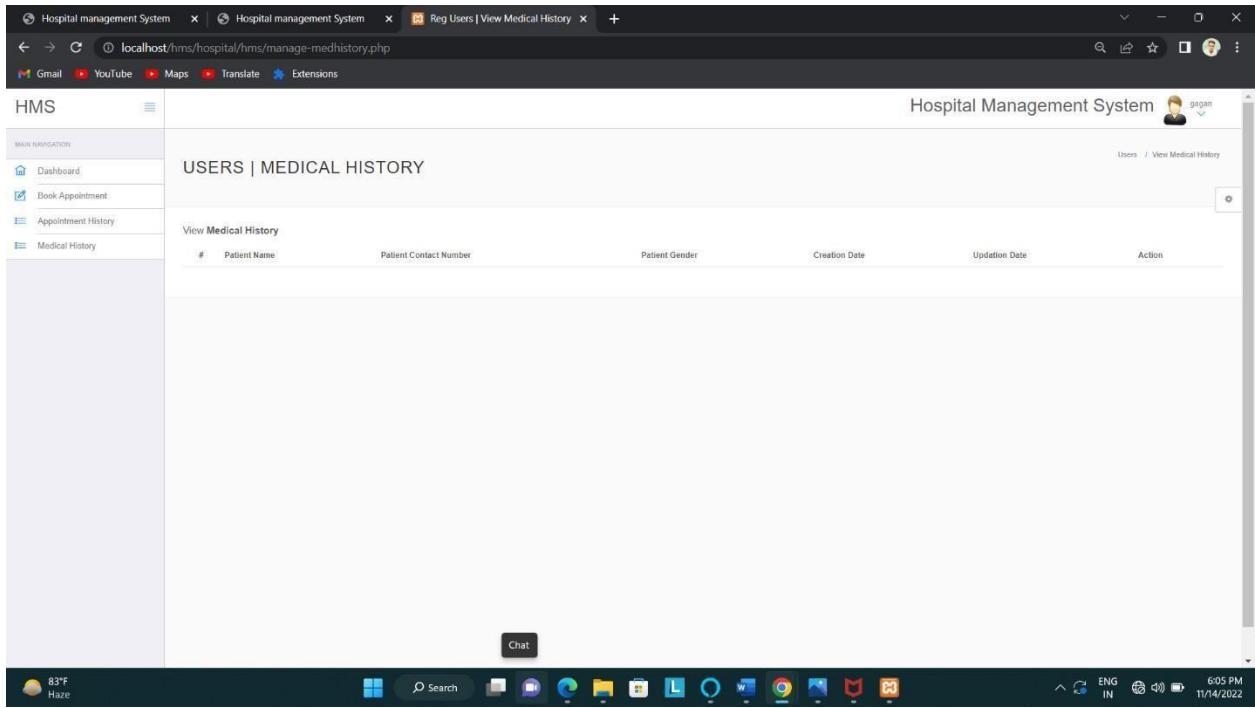
Top Screenshot (Book Appointment Page):

- Header:** Hospital management System, User | Book Appointment, localhost/hms/hospital/hms/book-appointment.php
- Main Navigation:** Dashboard, Book Appointment (selected), Appointment History, Medical History
- Title:** USER | BOOK APPOINTMENT
- Form Fields:**
 - Doctor Specialization: Select Specialization
 - Doctors: Select Doctor
 - Consultancy Fees: [Input field]
 - Date: [Input field]
 - Time: 6:00 PM eg : 10:00 PM
- Buttons:** Submit

Bottom Screenshot (Appointment History Page):

- Header:** Hospital management System, User | Appointment History, localhost/hms/hospital/hms/appointment-history.php
- Main Navigation:** Dashboard, Book Appointment, Appointment History (selected), Medical History
- Title:** USER | APPOINTMENT HISTORY
- Table:** A table showing appointment history details.

#	Doctor Name	Specialization	Consultancy Fee	Appointment Date / Time	Appointment Creation Date	Current Status	Action
1	ravi pandro	Dental Care	700	2022-11-23 / 12:15 PM	2022-11-11 12:07:39	Active	Cancel



Flow:

(1) User Logs in with user email and password.

Alternate Flow: (1) If the user email is wrong then it is asked to login again.

(2) If the password is wrong then the user is asked to enter again.

Doctor login page

Actor: User

HMS | Doctor Login

Sign in to your account
Please enter your name and password to log in.

Username

Password

This field is required.

[Forgot Password ?](#)

HOSPITAL MANAGEMENT SYSTEM

Hospital Management System ravi pande

Doctor / Appointment History

DOCTOR | APPOINTMENT HISTORY

#	Patient Name	Specialization	Consultancy Fee	Appointment Date / Time	Appointment Creation Date	Current Status	Action
1.	dalveer	Dental Care	700	2022-11-14 / 4:59 PM	2022-11-14 17:58:07	Canceled by you	Canceled
2.	gagan	Dental Care	700	2022-11-23 / 12:15 PM	2022-11-11 12:07:39	Active	Cancel
3.	manish	Dental Care	700	2022-11-15 / 1:30 PM	2022-11-11 12:25:43	Active	Cancel
4.	prince kumar	Dental Care	700	2022-11-21 / 2:15 PM	2022-11-11 14:06:39	Active	Cancel
5.	suraj jain	Dental Care	700	2022-11-22 / 2:15 PM	2022-11-11 14:11:58	Active	Cancel

HOSPITAL MANAGEMENT SYSTEM

Hospital Management System

localhost/hms/hospital/hms/doctor/view-patient.php?viewid=3

Gmail YouTube Maps Translate Extensions

HMS

MAIN NAVIGATION

- Dashboard
- Appointment History
- Patients
- Search

Hospital Management System ravi pandre

Doctor / Manage Patients

DOCTOR | MANAGE PATIENTS

Manage Patients

Patient Details			
Patient Name	prince kumar	Patient Email	princekumar2@gmail.com
Patient Mobile Number	7893430429	Patient Address	teen patti jabalpur
Patient Gender	male	Patient Age	30
Patient Medical History(if any)	sugar, asthma	Patient Reg Date	2022-11-14 18:01:26

Medical History					
#	Blood Pressure	Weight	Blood Sugar	Body Temperature	Medical Prescription
					Add Medical History

HOSPITAL MANAGEMENT SYSTEM

83°F Sunny

3:27 PM 11/15/2022

Hospital Management System

localhost/hms/hospital/hms/doctor/manage-patient.php

Gmail YouTube Maps Translate Extensions

HMS

MAIN NAVIGATION

- Dashboard
- Appointment History
- Patients
- Search

Hospital Management System ravi pandre

Doctor / Manage Patients

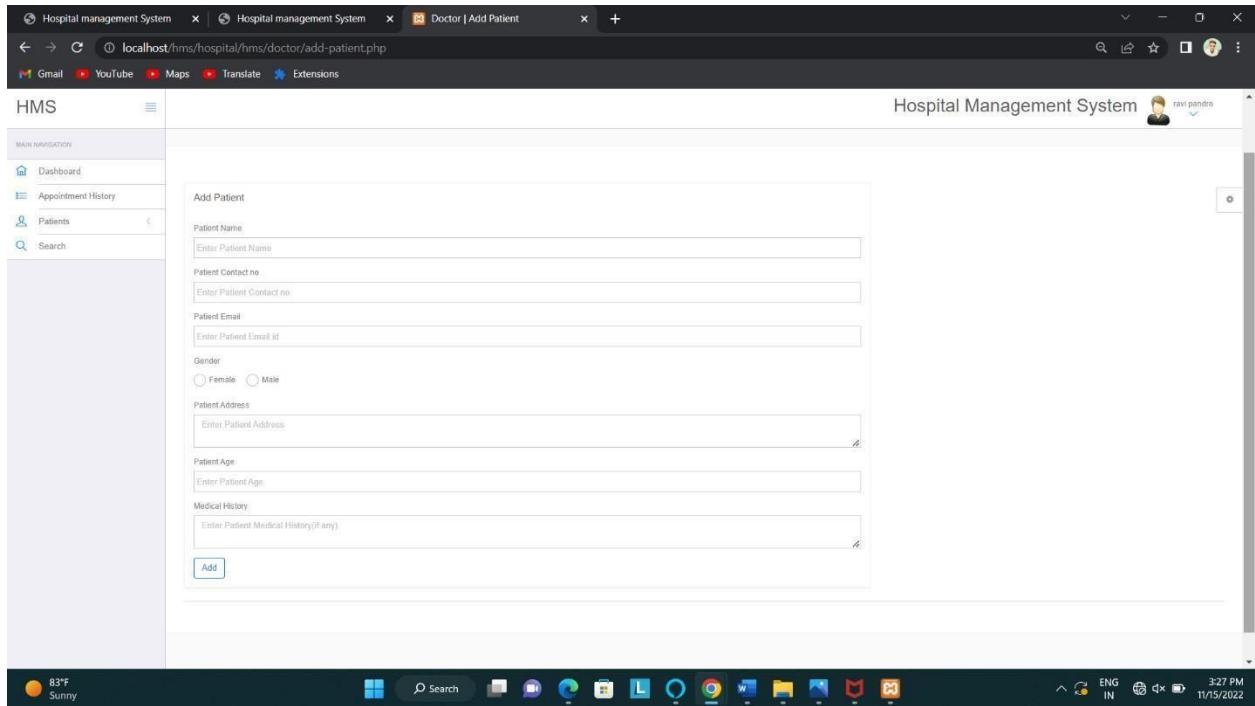
DOCTOR | MANAGE PATIENTS

Manage Patients

#	Patient Name	Patient Contact Number	Patient Gender	Creation Date	Updation Date	Action
1:	prince kumar	7893430429	male	2022-11-14 18:01:26		Edit Delete

82°F Sunny

3:27 PM 11/15/2022

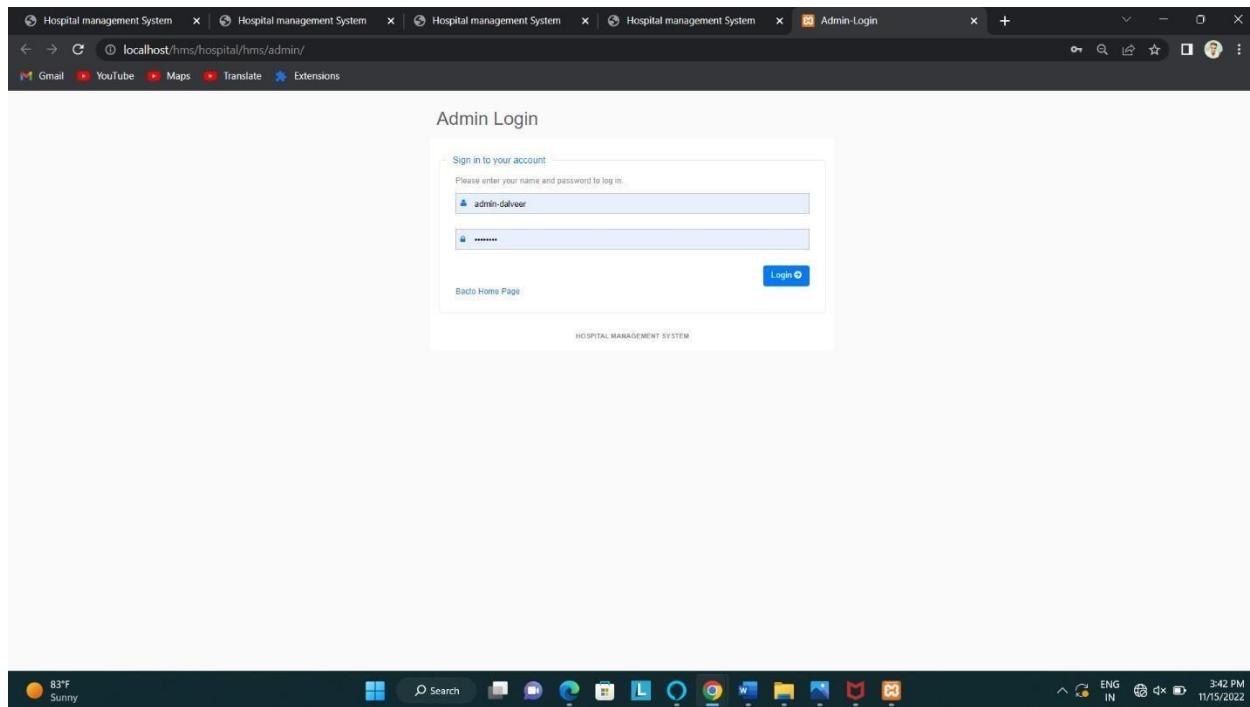


Flow: (1) User Logs in with user email and password.

Admin login

Actor: Admin users

Output: Admin details page



Flow: 1. Admin user can browse this page.

2. Admin user can view

Hospital Management System Admin | View Patients

localhost/hms/hospital/hms/admin/manage-patient.php

MAIN NAVIGATION

- Dashboard
- Doctors
- Users
- Patients
- Appointment History
- Consultus Queries
- Doctor Session Logs
- User Session Logs
- Reports
- Pages
- Patient Search

ADMIN | VIEW PATIENTS

View Patients

#	Patient Name	Patient Contact Number	Patient Gender	Creation Date	Updation Date	Action
1.	Amit Kumar	1231231230	male	2022-11-06 10:40:31		
2.	prince kumar	7893430429	male	2022-11-11 14:03:16		
3.	prince kumar	7893430429	male	2022-11-14 18:01:26		

Hospital Management System Admin

localhost/hms/hospital/hms/admin/appointment-history.php

MAIN NAVIGATION

- Dashboard
- Doctors
- Users
- Patients
- Appointment History
- Consultus Queries
- Doctor Session Logs
- User Session Logs
- Reports
- Pages
- Patient Search

PATIENTS | APPOINTMENT HISTORY

#	Doctor Name	Patient Name	Specialization	Consultancy Fee	Appointment Date / Time	Appointment Creation Date	Current Status	Action
1.	ravi pandro	dalever	Dental Care	700	2022-11-14 / 4:59 PM	2022-11-14 17:58:07	Active	No Action yet
2.	ravi pandro	gagan	Dental Care	700	2022-11-23 / 12:15 PM	2022-11-11 12:07:39	Active	No Action yet
3.	ravi pandro	manish	Dental Care	700	2022-11-15 / 1:30 PM	2022-11-11 12:25:43	Active	No Action yet
4.	prasant kumar	kamal	Neurologists	25000	2022-11-20 / 2:00 PM	2022-11-11 13:49:09	Active	No Action yet
5.	sandeep sharma	prince kumar	covid -19	250	2022-11-25 / 2:15 PM	2022-11-11 14:05:36	Active	No Action yet
6.	Dr bangali	prince kumar	Obstetrics and Gynecology	200	2022-11-28 / 2:15 PM	2022-11-11 14:06:02	Active	No Action yet
7.	neetu yadav	prince kumar	Pediatrics	800	2022-11-30 / 2:15 PM	2022-11-11 14:06:19	Active	No Action yet
8.	ravi pandro	prince kumar	Dental Care	700	2022-11-21 / 2:15 PM	2022-11-11 14:06:39	Active	No Action yet
9.	pushpraj bhatle	suraj jain	Radiology	500	2022-11-29 / 2:15 PM	2022-11-11 14:11:19	Active	No Action yet
10.	amit jayaram	suraj jain	Orthopedics	500	2022-11-28 / 2:15 PM	2022-11-11 14:11:38	Active	No Action yet
11.	ravi pandro	suraj jain	Dental Care	700	2022-11-22 / 2:15 PM	2022-11-11 14:11:58	Active	No Action yet
12.	aditya pathak	suraj jain	General Surgery	15000	2022-11-23 / 2:15 PM	2022-11-11 14:12:18	Active	No Action yet
13.	sourabh joshi	suraj jain	Ophthalmology	300	2022-11-23 / 2:15 PM	2022-11-11 14:12:37	Active	No Action yet

all module here

Hospital Management System

localhost/hms/hospital/hms/admin/manage-doctors.php

Gmail YouTube Maps Translate Extensions

HMS

MAIN NAVIGATION

- Dashboard
- Doctors
- Users
- Patients
- Appointment History
- Consultus Queries
- Doctor Session Logs
- User Session Logs
- Reports
- Pages
- Patient Search

Hospital Management System Admin

ADMIN | MANAGE DOCTORS

Manage Doctors

#	Specialization	Doctor Name	Creation Date	Action
1.	ENT	Anuj Kumar	2022-10-30 23:46:52	
2.	Endocrinologists	Charu Dua	2022-11-04 06:38:41	
3.	Dental Care	Ravi Pandre	2022-11-11 12:01:13	
4.	covid-19	Sandeep Sharma	2022-11-11 12:40:48	
5.	General Surgery	Aditya Pottakal	2022-11-11 12:44:33	
6.	Pediatrics	Nehru Yadav	2022-11-11 12:50:53	
7.	Anesthesia	Himanshu Lukharni	2022-11-11 12:57:31	
8.	Neurologists	Prasant Kumar	2022-11-11 13:04:34	
9.	Radiology	Pushpraj Bhatele	2022-11-11 13:09:23	
10.	Ophthalmology	Sourabh Joshi	2022-11-11 13:15:17	
11.	Dermatologists	Sneha Tripathi	2022-11-11 13:19:56	
12.	Endocrinologists	Aparna Soni	2022-11-11 13:23:55	
13.	Child Specialist	B.D Sahu	2022-11-11 13:29:05	
14.	Obstetrics and Gynecology	Dr Bangali	2022-11-11 13:34:15	
15.	Orthopedics	Amil Jaykumar	2022-11-11 13:38:58	

83°F Sunny 3:43 PM 11/15/2022

Hospital Management System Admin

ADMIN | MANAGE USERS

Manage Users

#	Full Name	Address	City	Gender	Email	Creation Date	Updation Date	Action
1.	Dalveer	Jabalpur	Jabalpur	Male	dalveersinghbilagar@gmail.com	2022-11-11 11:51:55		
2.	Dalveer	Jabalpur	Jabalpur	Male	dalveer@gmail.com	2022-11-11 11:53:43		
3.	Gagan	Peenedy	Jabalpur	Male	gagan@gmail.com	2022-11-11 12:05:14		
4.	Manish	Giranghat	Jabalpur	Male	manish@gmail.com	2022-11-11 12:23:28		
5.	Kamal	Bilari	Jabalpur	Male	kamal@gmail.com	2022-11-11 13:47:49		
6.	Prince Kumar	Bilari	Jabalpur	Male	princekumar@gmail.com	2022-11-11 14:05:05		
7.	Suraj Jain	Cadar	Jabalpur	Male	surajjain@gmail.com	2022-11-11 14:10:45		

82°F Sunny 3:43 PM 11/15/2022

Hospital Management System

localhost/hms/hospital/hms/admin/user-logs.php

HMS

MAIN NAVIGATION

- Dashboard
- Doctors
- Users
- Patients
- Appointment History
- Consultus Queries
- Doctor Session Logs
- User Session Logs
- Reports
- Pages
- Patient Search

Hospital Management System Admin

ADMIN | USER SESSION LOGS

#	User id	Username	User IP	Login time	Logout Time	Status
1.	1	john doe12@test.com	::1	2022-11-06 17:44:11		Success
2.	1	john doe12@test.com	::1	2022-11-06 17:51:29	06-11-2022 05:53:00 PM	Success
3.		amit@gmail.com	::1	2022-11-06 18:45:43		Failed
4.	2	amit@gmail.com	::1	2022-11-06 18:45:58	06-11-2022 06:50:48 PM	Success
5.		anujk123@test.com	::1	2022-11-11 10:50:49		Failed
6.		admin	::1	2022-11-11 11:50:50		Failed
7.		admin	::1	2022-11-11 11:50:51		Failed
8.		admin	::1	2022-11-11 11:50:52		Failed
9.		admin	::1	2022-11-11 11:50:52		Failed
10.		admin	::1	2022-11-11 11:50:53		Failed
11.		admin	::1	2022-11-11 11:50:53		Failed
12.	4	daiveer@gmail.com	::1	2022-11-11 11:54:23	14-11-2022 06:04:24 PM	Success
13.	5	gagan@gmail.com	::1	2022-11-11 12:06:31		Success
14.	5	gagan@gmail.com	::1	2022-11-11 12:11:58	11-11-2022 12:13:21 PM	Success
15.	6	march@gmail.com	::1	2022-11-11 12:23:58	11-11-2022 12:25:59 PM	Success
16.	7	kamal@gmail.com	::1	2022-11-11 13:48:18	11-11-2022 01:49:41 PM	Success
17.	8	princerumar@gmail.com	::1	2022-11-11 14:05:18	11-11-2022 02:05:47 PM	Success
18.	9	surajin@gmail.com	::1	2022-11-11 14:10:56		Success
19.	5	gagan@gmail.com	::1	2022-11-14 17:53:13		Success

83°F Sunny

ENG IN 3:44 PM 11/15/2022

Hospital Management System

localhost/hms/hospital/hms/admin/doctor-logs.php

HMS

MAIN NAVIGATION

- Dashboard
- Doctors
- Users
- Patients
- Appointment History
- Consultus Queries
- Doctor Session Logs
- User Session Logs
- Reports
- Pages
- Patient Search

Hospital Management System Admin

ADMIN | DOCTOR SESSION LOGS

#	User id	Username	User IP	Login time	Logout Time	Status
1.	1	gfgdf	::1	2022-11-04 08:32:16		Failed
2.	2	charudua12@test.com	::1	2022-11-06 11:29:40	06-11-2022 05:35:10 PM	Success
3.	2	charudua12@test.com	::1	2022-11-06 17:36:37	06-11-2022 05:36:40 PM	Success
4.	2	charudua12@test.com	::1	2022-11-06 17:38:56	06-11-2022 05:42:53 PM	Success
5.	1	anujk123@test.com	::1	2022-11-06 17:53:18	06-11-2022 05:53:40 PM	Success
6.	2	charudua12@test.com	::1	2022-11-06 18:46:53	06-11-2022 06:47:07 PM	Success
7.	1	anujk123@test.com	::1	2022-11-06 18:47:33	06-11-2022 06:59:28 PM	Success
8.		john doe12@test.com	::1	2022-11-11 11:49:12		Failed
9.		john doe12@test.com	::1	2022-11-11 11:49:48		Failed
10.		admin	::1	2022-11-11 11:49:50		Failed
11.		admin	::1	2022-11-11 11:49:51		Failed
12.		admin	::1	2022-11-11 11:49:52		Failed
13.		admin	::1	2022-11-11 11:49:53		Failed
14.		admin	::1	2022-11-11 11:49:53		Failed
15.		admin	::1	2022-11-11 11:49:54		Failed
16.		admin	::1	2022-11-11 11:49:54		Failed
17.		admin	::1	2022-11-11 11:49:55		Failed
18.		admin	::1	2022-11-11 11:49:55		Failed
19.	4	ravi@gmail.com	::1	2022-11-11 12:15:01	11-11-2022 12:16:23 PM	Success

82°F Sunny

ENG IN 3:44 PM 11/15/2022

The screenshot shows the 'Admin | Query Details' page of the Hospital Management System. The URL in the address bar is `localhost/hms/hospital/hms/admin/query-details.php?id=1`. The page title is 'Hospital Management System'. On the left, there is a sidebar titled 'HMS' with a 'MAIN NAVIGATION' section containing links for Dashboard, Doctors, Users, Patients, Appointment History, Comatus Queries, Doctor Session Logs, User Session Logs, Reports, Pages, and Patient Search. The main content area is titled 'ADMIN | QUERY DETAILS' and contains a form titled 'Manage Query Details'. The form fields include: Full Name (Anuj kumar), Email Id (anuj30@test.com), Contact Number (1425362514), Message (This is for testing purposes. This is for testing purposes.), Query Date (2022-10-30 22:22:03), and Admin Remark (empty). A blue 'Update' button is at the bottom. The status bar at the bottom shows the weather as 83°F Sunny.

The screenshot shows the 'Admin | Dashboard' page of the Hospital Management System. The URL in the address bar is `localhost/hms/hospital/hms/admin/dashboard.php`. The page title is 'Hospital Management System'. The sidebar on the left is identical to the one in the previous screenshot. The main content area is titled 'ADMIN | DASHBOARD' and features five cards: 'Manage Users' (Total Users: 7), 'Manage Doctors' (Total Doctors: 15), 'Appointments' (Total Appointments: 17), 'Manage Patients' (Total Patients: 3), and 'New Queries' (Total New Queries: 1). The status bar at the bottom shows the weather as 83°F Sunny.

Actor : Admin users

Output: Appointment History.

7. Conclusions

This project has been a rewarding experience in more than one way. The entire project work has enlightened us in the following areas.

- a) We have gained an insight into the working of the HOSPITAL. This represents a typical real world situation.
- b) Our understanding of database design has been strengthened this is because in order to generate the final reports of database designing has to be properly followed.
- c) Scheduling a project and adhering to that schedule creates a strong sense of time management.
- d) Sense of teamwork has developed and confidence of handling real life project has increased to a great extent.
- e) Initially, there were problem with the validation but with discussions, we were to implement validations.

Limitations of the system:- Online payment is not available at this version.

 Data delete & edit system is not available for all section.

 User account not verified by Mobile SMS not available in this system.

 Loss of data due to mismanagement.

Future plan :-Diagnostics billing system.

8. References

- [1] Deepak Thomas "Beginning PHP 4 Databases", Wrox Press Ltd. Paperback-17, October, 2002.70-130 pp.
- [2] Matt Doyle, "Beginning PHP 5.3, 2ndedition", October 2009. 150-270 pp.
- [3] Luke Welling, Laura Thomson. Sams PHP and MySQL Web Development, 2nd edition, Paperback- 20 February, 2003. 105-209 pp.
- [4] W. Jason Gilmore "Beginning PHP 5 and MySQL 5 from Novice to Professional SECOND EDITION", Jul 9, 2008.100-150 pp.

- [5] Abraham Silberschatz, Henry F. Korth and S. Sudarshan "Sixth Edition Database System Concepts released", January 28, 2010. 206-253 pp.
- [6] Server-Side Scripting <http://php.net/manual/en/index.php>, Last accessed on 8/15/2022 at 2:33pm
- [7] HTML & CSS <https://www.w3schools.com/>, Last accessed on 10/2/2022 at 1:33pm.
- [8] Bootstrap <http://getbootstrap.com/>, last accessed on 09/30/2022 at 10:10pm.
- [9] <https://stackoverflow.com/>, last accessed on 11/07/2022 at 2:20am.

9.GITHUB LINK (FOR PROJECT REPORT AND PROJECT CODE)

LINK:-

<https://github.com/Himanshu95841775/Hospital-management-system-.git>