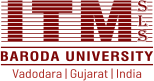
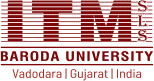
**Project Report On**

**Pharmacy-Drug-Management-System project**



By

**Student Name**

*In partial fulfillment of requirements for the award of the degree*

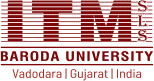
**Bachelor of Technology (CSE)**

*Under the guidance of*

*Aadash S Patel*

**College Name**

**ITM(sls) Baroda University**



**ITM(sls) Baroda University**

**Department of**

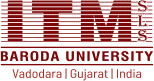
**Computer science**

**Certificate**

This is to certify that Tanmay Jadav (20C21017), student of Bachelor of Technology, VI Semester, Department of Computer Science of ITM(sls) Baroda university, has pursued the Project titled “Pharmacy-Drug-Management-System project” under the supervision of Aadarsh S. Patel and the report has been submitted in partial fulfillment of requirements for the award of the degree, Bachelor of Technology in Computer Science ITM(sls) Baroda university in the Year 2022.

Supervisor

Aadash S Patel



**ITM(sls) Baroda university**

**Department of**

**Computer Science**

**Certificate**

This is to certify that Tanmay Jadav(20C21017), student of Bachelor of Technology, VI Semester, Department of Computer science of Institute of Technology and management (sls) Baroda University, has pursued the Major Project titled “Pharmacy-Drug-Management-System project” under the supervision Aadarsh S. Patel and the report has been submitted in partial fulfillment of requirements for the award of the degree, Bachelor of Technology Computer Science ITM(sls) Baroda university in the Year 2022.

Candidate

Signature

Examiner

Signature

**ACKNOWLEDGEMENT**

I express my sincere regard and indebtedness to my project external guide Aadash S. Patel for his valuable time, guidance, encouragement, support and cooperation throughout the duration of our project. I would sincerely like to thank IT Department for giving me the opportunity to work on enhancing my technical skills while undergoing this project. This project was done under the guidance of Adarsh Patel. This project helped in understanding the various parameters which are involved in the development of a web application and the working and integration of front end along with the back end to create a fully functional web application.

Student Name

TANMAY JADAV

Roll Number

20C21017

**ABSTRACT**

This project is a web based shopping system for an existing Pharmacy.The project objective is to deliver the online medicine shopping application website This project is an attempt to provide the advantages of online medicine shopping to customers of a real Pharmacy. It helps buying the products in the Pharmacy any where through internet by using an android device. Thus the customer will get the service of online shopping and home delivery from his favorite Pharmacy store. This system can be implemented to any pharmacy store in the locality or to multi national branded Pharmacy store having retail outlet chains. If Pharmacy store are providing an online portal where their customers can enjoy easy shopping from anywhere, the Pharmacy store won’t be losing any more customers to the trending online Pharmacy store such as 1MG or medcart. Since the application is available in the Smartphone it is easily accessible and always available.

Online medicine shopping is the process whereby consumers directly buy goods or services from a seller in real-time, without an intermediary service, over the Internet. It is a form of electronic commerce. This project is an attempt to provide the advantages of online medicine shopping to customers of a real shop.It helps buying the products in the shop anywhere through internet by using an web app on any device. Thus the customer will get the service of online shopping and home delivery from his Pharmacy store

**CHAPTER 1 : INTRODUCTION**

**1.1 INTRODUCTION**

The project, Pharmacy store Management System is a web-based application that allows the Pharmacy store manager to handle all Pharmacy store activities online. Interactive GUI and the ability to manage various shopping stores make this system very flexible and convenient. The shop manager is a very busy person and does not have the time to sit and manage the entire activities manually on paper. This web application gives him the power and flexibility to manage the entire system from a single online system. The system allows the manager to post available items in the system. Customers can view and shop online. Admin has the power of either approving or disapproving the customer’s booking request. The system is hence useful for both customers and managers to por items manage the shopping activities.

**1.2 AIM**

The mission is to facilitate easy management and administration of a small business like Pharmacy store with capabilities to do Booking or shopping of the items, Cancellation of the items, Cash billing, service,return service, Billing, Total Billing, Travels arrangement etc. using the automated shopping web application. One can Keep detailed records or info on an unlimited amount of customers. The system lets the user Know which all items are available for occupancy at any point of time. This makes the Booking considerably faster. And thus helps the Pharmacy store in better management and reduce a lot of paper work as well as manpower.

**1.3 EXISTING SYSTEM**

In the existing system, only provides the information about particular Pharmacy store and only some of the Pharmacy stores has possibility to reserve the items. The previous system was failure to publish the items’s availability of multiple Pharmacy stores. And, it is failure to provide defence to admin and user information.

**1.4 PROPOSED SYSTEM**

The Pharmacy store management system provides the quality service to the end user. This project aims at creating on Pharmacy store Management System which can be used by Admin and Customers. The admin to advise/publish the availability of itemss in different Pharmacy stores and customers are checking the availability of items in required Pharmacy store. Customers should be able to know the availability of the itemss on a particular date to reserve in Pharmacy store. They should be able to reserve the available itemss according to their need in advance. To make their stay comforitems. The Admin hands the booking information of customers. The users can register and log into the system. The administrator will know the details of revnue and daily income.

**1.5 FEASIBILITY STUDY**

A feasibility study is a high-level capsule version of the entire System analysis and Design Process. The study begins by classifying the problem definition. Feasibility is to determine if it’s worth doing. Once an acceptance problem definition has been generated, the analyst develops a logical model of the system. A search for alternatives is analyzed carefully. There are 3 parts in feasibility study.

**1.5.1 OPERATIONAL FEASIBILITY**

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes.To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, producibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviours are to be realised. A system design and development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases.

**1.5.2 TECHNICAL FEASIBILITY**

This involves questions such as whether the technology needed for the system exists, how difficult it will be to build, and whether the firm has enough experience using that technology. The assessment is based on outline design of system requirements in terms of input, processes, output, fields, programs and procedures. This can be qualified in terms of volume of data, trends, frequency of updating inorder to give an introduction to the technical system. The application is the fact that it has been developed on windows XP platform and a high configuration of 1GB RAM on Intel Pentium Dual core processor. This is technically feasible .The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system.

**1.5.3 ECONOMICAL FEASIBILITY**

Establishing the cost-effectiveness of the proposed system i.e. if the benefits do not outweigh the costs then it is not worth going ahead. In the fast paced world today there is a great need of online social networking facilities. Thus the benefits of this project in the current scenario make it economically feasible. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/benefits analysis.

**CHAPTER 2 : SOFTWARE REQUIREMENTS SPECIFICATION**

**2.1 Hardware Requirements**

|  |  |
| --- | --- |
| **Number** | **Description** |
| 1 | PC with 250 GB or more Hard disk. |
| 2 | PC with 2 GB RAM. |
| 3 | PC with Pentium 1 and Above. |

**2.2 Software Requirements**

|  |  |  |
| --- | --- | --- |
| **Number** | **Description** | **Type** |
| 1 | Operating System | Windows XP / Windows |
| 2 | Language | Java Script Node Js |
| 3 | Database | MySQL |
| 4 | IDE | Visual Code |
| 5 | Browser | Google Chrome |

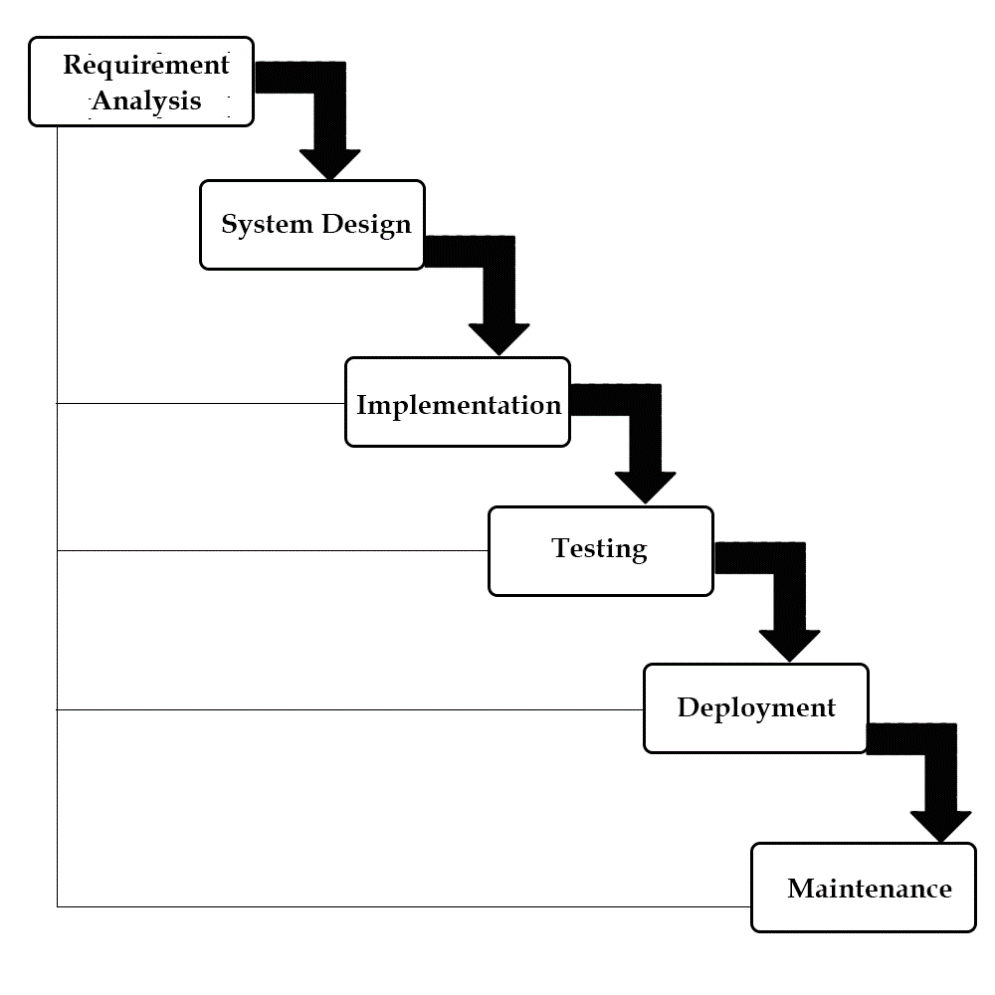
**CHAPTER 3 : DESIGN & PLANNING**

**3.1 Software Development Life Cycle Model**

**3.1.1 WATERFALL MODEL**

The waterfall model was selected as the SDLC model due to the following reasons:

* Requirements were very well documented, clear and fixed.
* Technology was adequately understood.
* Simple and easy to understand and use.
* There were no ambiguous requirements.
* Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
* Clearly defined stages.
* Well understood milestones.Easy to arrange tasks.



**CHAPTER 4 : IMPLEMENTATION DETAILS**

**4.1 : FRONT END**

**4.1.1 HTML    **

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <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 such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

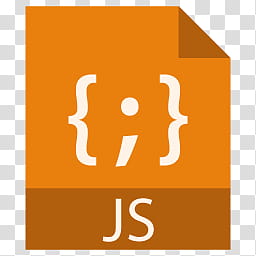
**4.1.2 Css    **

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML.CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts.This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

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. The 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.

**4.1.3 JavaScript    **

JavaScript s a high-level, interpreted scripting language that conforms to the ECMAScript specification. JavaScript has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web.JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it,and major web browsers have a dedicated JavaScript engine to execute it.As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has APIs for working with text, arrays, dates, regular expressions, and the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities. It relies upon the host environment in which it is embedded to provide these features.

Initially only implemented client-side in web browsers, JavaScript engines are now embedded in many other types of host software, including server-side in web servers and databases, and in non-web programs such as word processors and PDF software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets.

The terms Vanilla JavaScript and Vanilla JS refer to JavaScript not extended by any frameworks or additional libraries. Scripts written in Vanilla JS are plain JavaScript code.Google's Chrome extensions, Opera's extensions, Apple's Safari 5 extensions, Apple's Dashboard Widgets, Microsoft's Gadgets, Yahoo! Widgets, Google Desktop Gadgets, and Serence Klipfolio are implemented using JavaScript.

**4.2 :**

**4.2.1 Node Js    \\**

Node.js is an open-source server environment. Node.js is cross-platform and runs on Windows, Linux, Unix, and macOS. Node.js is a back-end JavaScript runtime environment. Node.js runs on the V8 JavaScript Engine and executes JavaScript code outside a web browser

Node.js lets developers use JavaScript to write command line tools and for server-side scripting. The functionality of running scripts server-side produces dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm,[6] unifying web-application development around a single programming language, rather than differnt languages for server-side and client-side scriptsNode.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in web applications with many input/output operations, as well as for real-time Web applications (e.g., real-time communication programs and browser games).[7]

The Node.js distributed development project was previously governed by the Node.js Foundation,[8] and has now merged with the JS Foundation to form the OpenJS Foundation. OpenJS Foundation is facilitated by the Linux Foundation's Collaborative Projects program.[9]

Node.js allows the creation of Web servers and networking tools using JavaScript and a collection of "modules" that handle various core functionalities.[ Modules are provided for file system I/O, networking (DNS, HTTP, TCP, TLS/SSL, or UDP), binary data (buffers), cryptography functions, data streams, and other core functions.[ Node.js's modules use an API designed to reduce the complexity of writing server applications.[

JavaScript is the only language that Node.js supports natively, but many compile-to-JS languages are available.[49] As a result, Node.js applications can be written in CoffeeScript,[50] Dart, TypeScript, ClojureScript and others.

Node.js is primarily used to build network programs such as Web servers.[45] The most significant difference between Node.js and PHP is that most functions in PHP block until completion (commands execute only after previous commands finish), while Node.js functions are non-blocking (commands execute concurrently or even in parallel,[51][52] and use callbacks to signal completion or failure).[45]

Node.js is officially supported on Linux, macOS and Microsoft Windows 8.1 and Server 2012 (and later),[3] with tier 2 support for SmartOS and IBM AIX and experimental support for FreeBSD. OpenBSD also works, and LTS versions available for IBM i (AS/400).[53] The provided source code may also be built on similar operating systems to those officially supported or be modified by third parties to support others such as NonStop OS[54] and Unix servers. such as generated HTML code or binary image data. PHP can be used for many programming tasks outside of the web context, such as standalone graphical applications and robotic drone control.

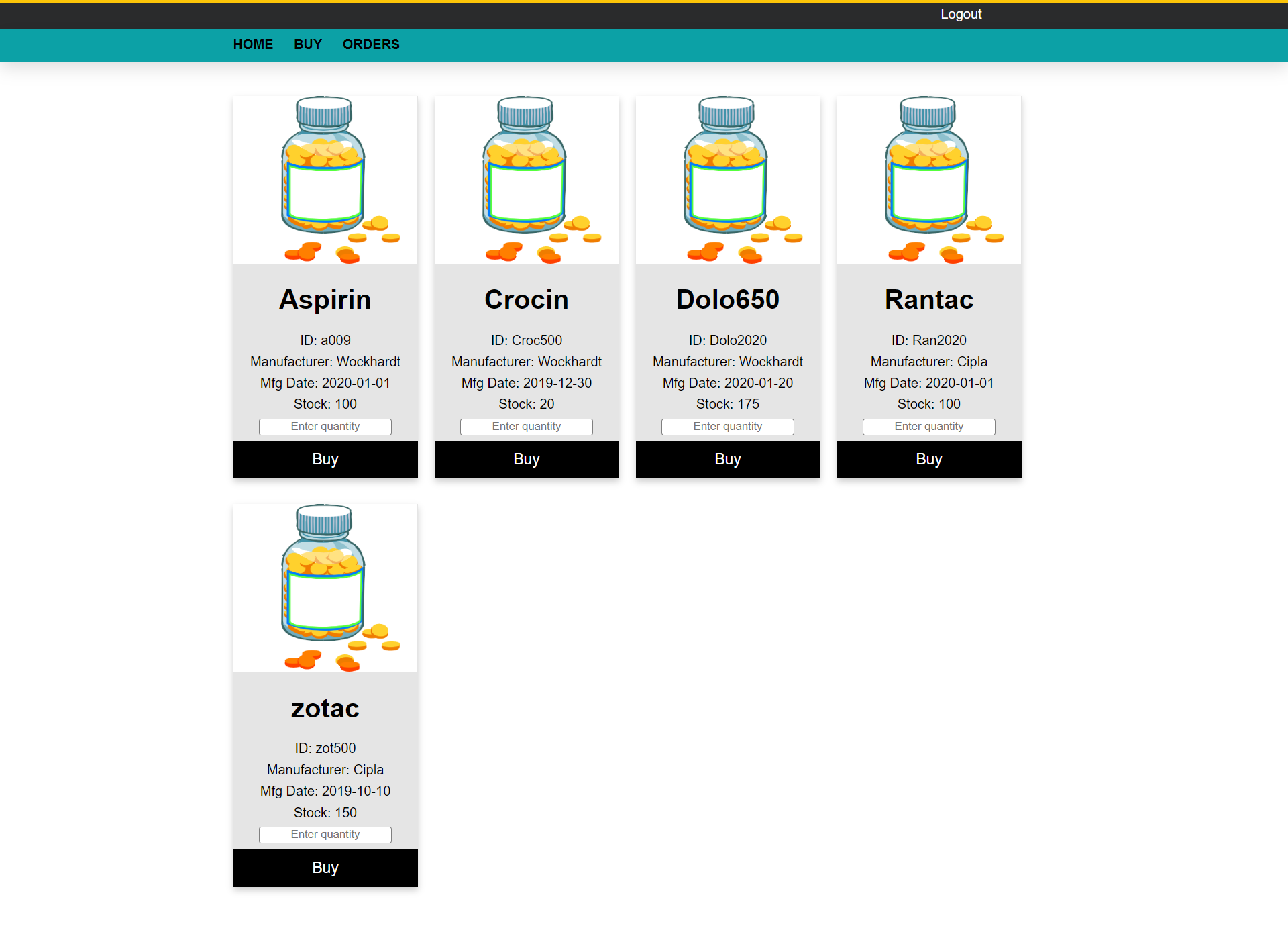
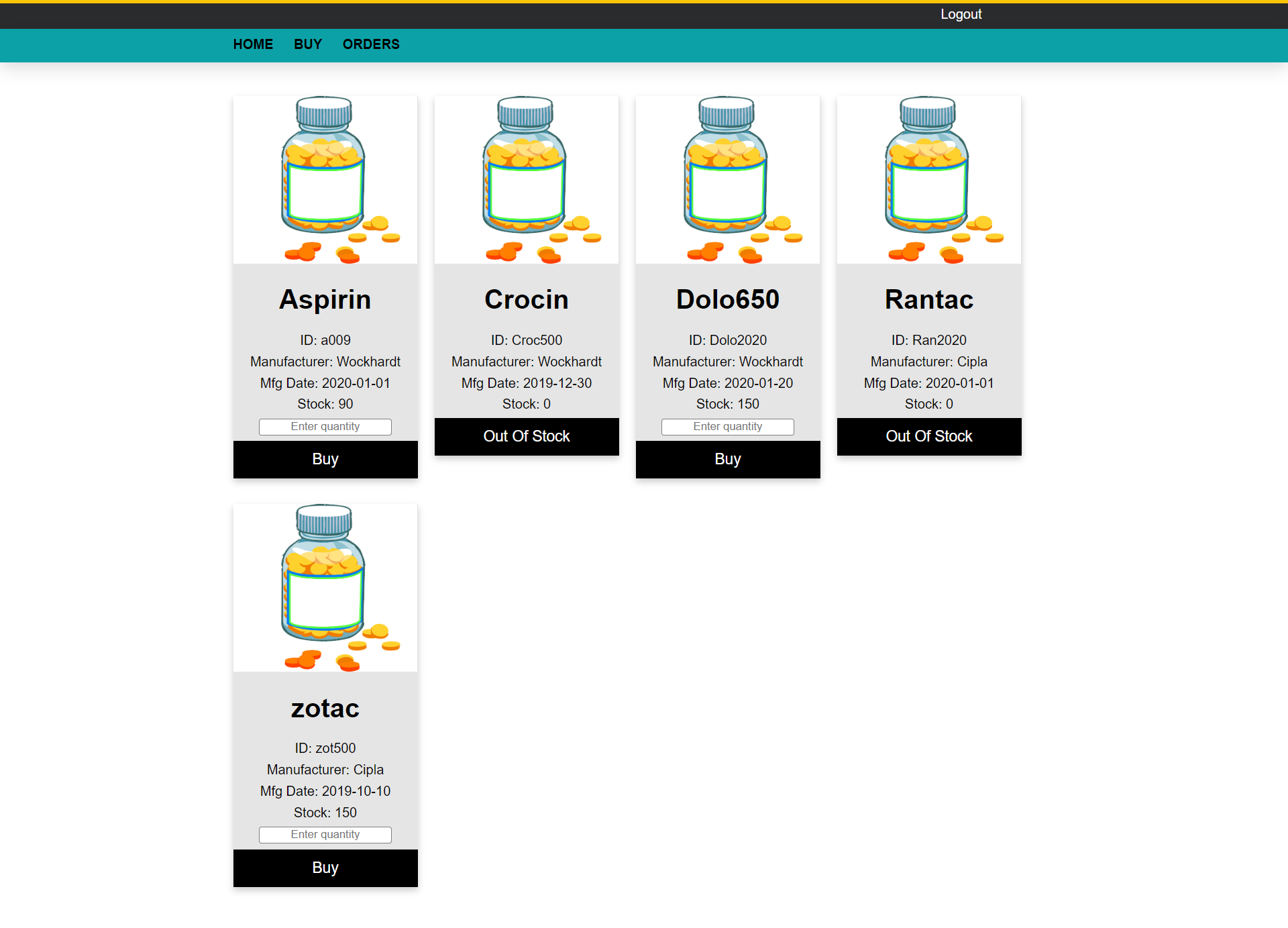
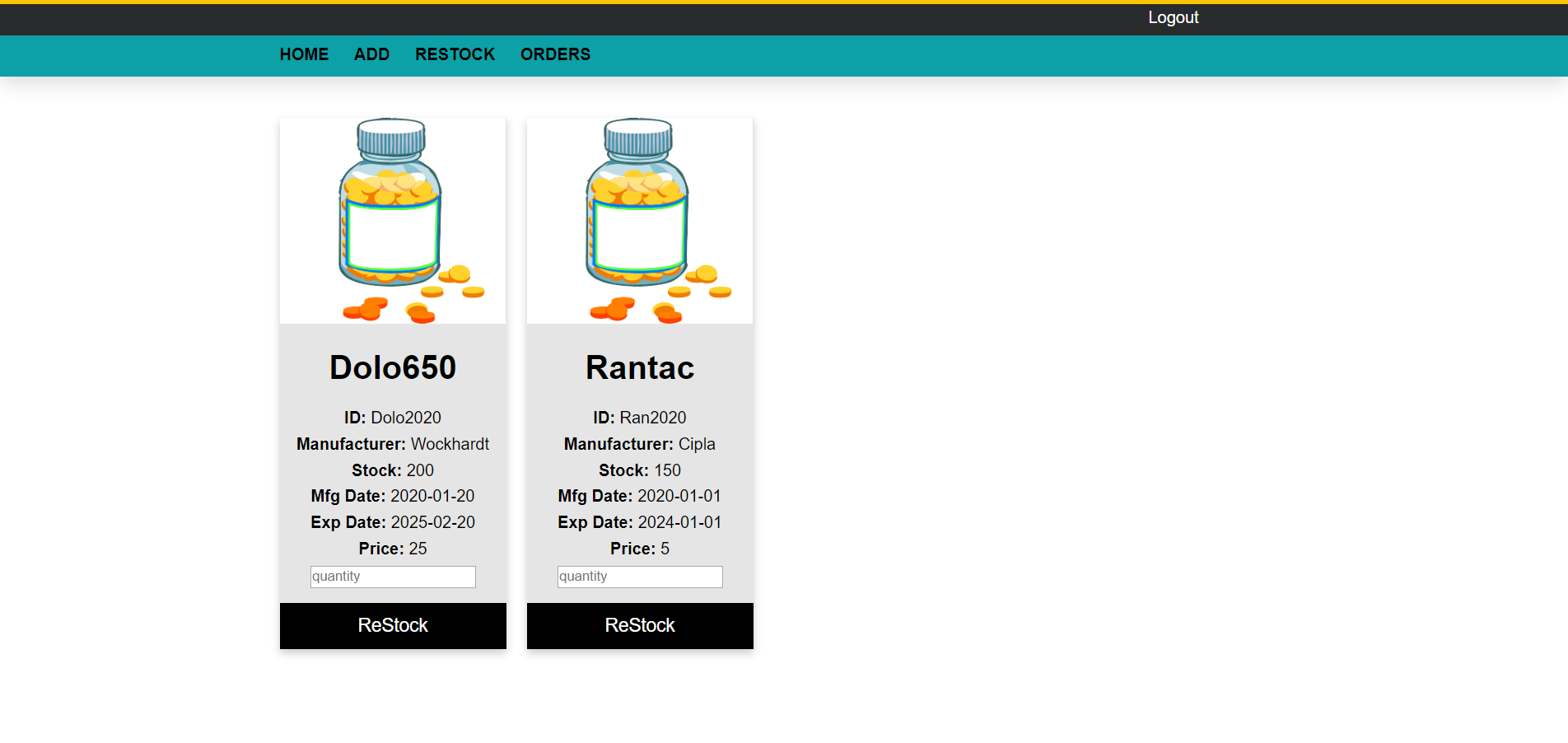
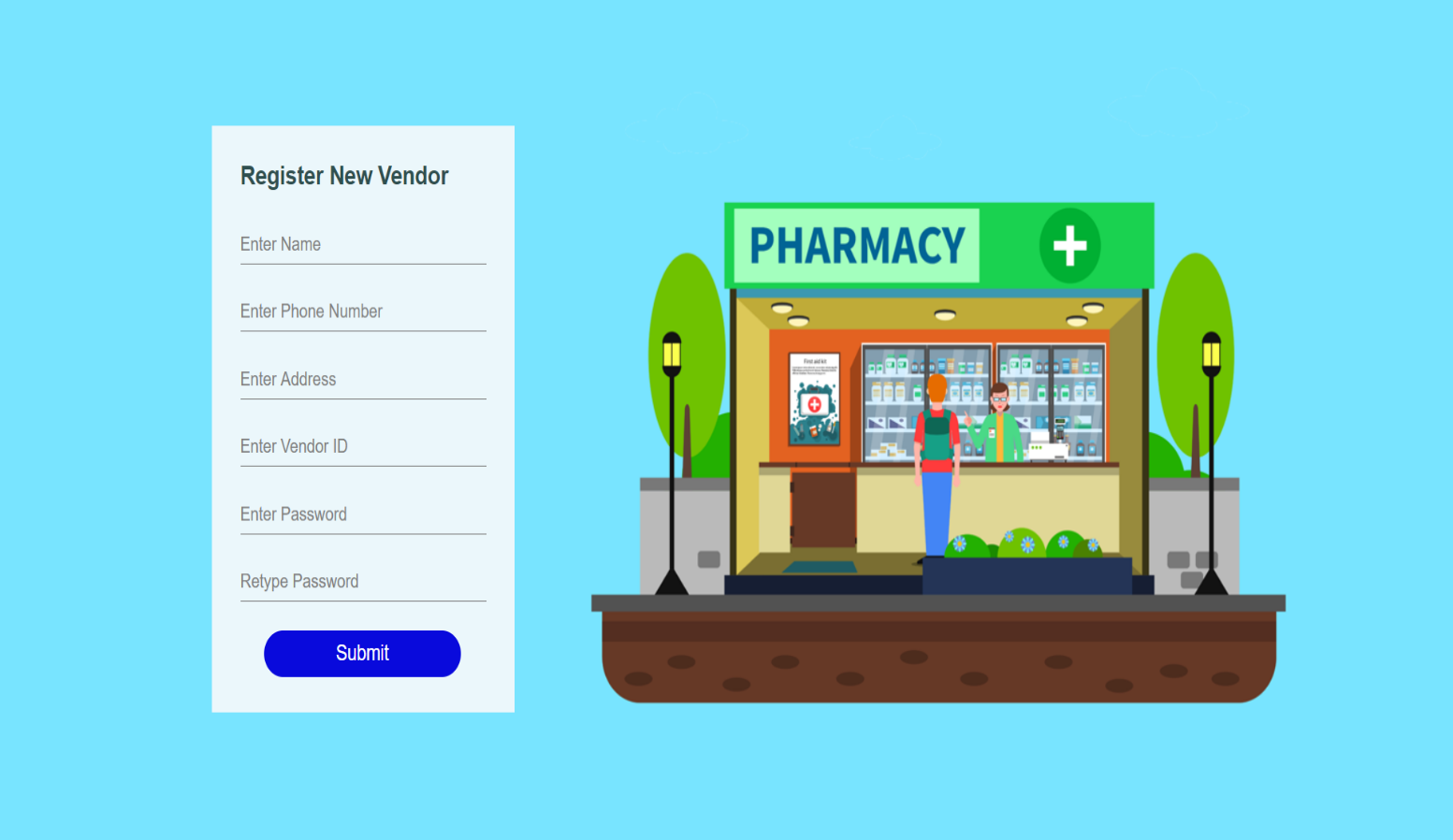
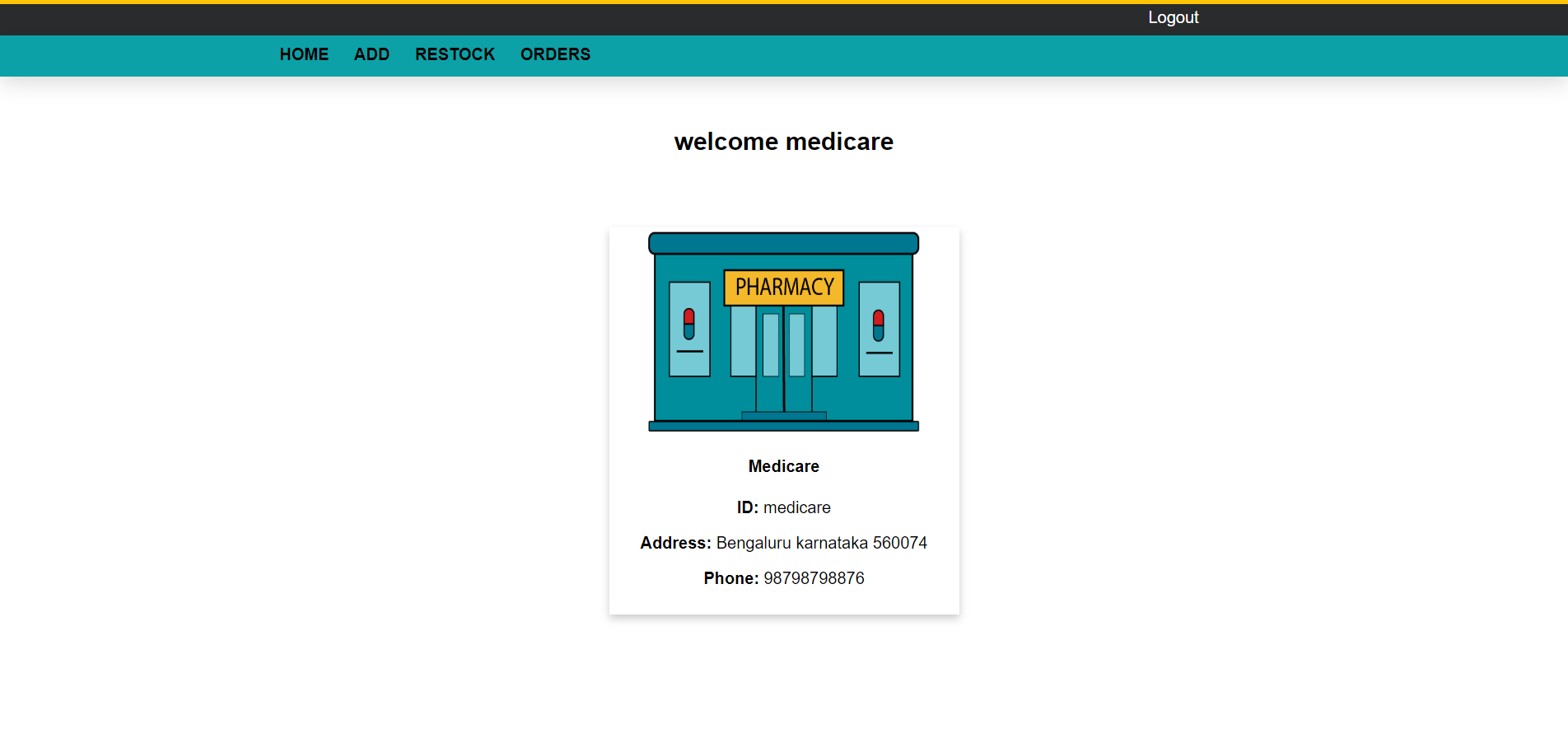
**4.2.2 MySQL    **

MySQL is an open source relational database management system (RDBMS) based on Structured Query Language (SQL). It is one part of the very popular LAMP platform consisting of Linux, Apache, My SQL, and PHP. Currently My SQL is owned by Oracle. My SQL database is available on most important OS platforms. It runs on BSD Unix, Linux, Windows, or Mac OS. Wikipedia and YouTube use My SQL. These sites manage millions of queries each day. My SQL comes in two versions: My SQL server system and My SQL embedded system.

RDBMS TERMINOLOGY

Before we proceed to explain MySQL database system, let's revise few definitions related to database.

* **Database:**A database is a collection of itemss, with related data.
* **Items:**A items is a matrix with data. A items in a database looks like a simple spadsheet.
* **Column:**One column (data element) contains data of one and the same kind, for example the column postcode.
* **Row:**A row (= tuple, entry or record) is a group of related data, for example the data of one subscription.
* **Redundancy:**Storing data twice, redundantly to make the system faster.
* **Primary Key:**A primary key is unique. A key value cannot occur twice in one items. With a key, you can find at most one row.
* **Foreign Key:**A foreign key is the linking pin between two itemss.
* **Compound Key:**A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.
* **Index:**An index in a database resembles an index at the back of a book.
* **Referential Integrity:**Referential Integrity makes sure that a foreign key value always points to an existing row.



**CHAPTER 5 : TESTING AND IMPLEMENTATION**

The term implementation has different meanings ranging from the conversation of a basic application to a complete replacement of a computer system. The procedures however, are virtually the same. Implementation includes all those activities that take place to convert from old system to new. The new system may be totally new replacing an existing manual or automated system or it may be major modification to an existing system. The method of implementation and time scale to be adopted is found out initially. Proper implementation is essential to provide a reliable system to meet organization requirement.

**5.1 : UNIT TESTING**

**5.1.1 Introduction**

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. Intuitively, one can view a unit as the smallest tesitems part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. It forms the basis for component testing. Ideally, each test case is independent from the others. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.

**5.1.2 Benifits**

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits.

**1) Find problems early :**Unit testing finds problems early in the development cycle. In test-driven development (TDD), which is frequently used in both extreme programming and scrum, unit tests are created before the code itself is written. When the tests pass, that code is considered complete. The same unit tests are run against that function frequently as the larger code base is developed either as the code is changed or via an automated process with the build. If the unit tests fail, it is considered to be a bug either in the changed code or the tests themselves. The unit tests then allow the location of the fault or failure to be easily traced. Since the unit tests alert the development team of the problem before handing the code off to testers or clients, it is still early in the development process.

**2 ) Facilitates Change :**Unit testing allows the programmer to refactor code or upgrade system libraries at a later date, and make sure the module still works correctly (e.g., in regression testing). The procedure is to write test cases for all functions and methods so that whenever a change causes a fault, it can be quickly identified. Unit tests detect changes which may break a design contract.

**3 ) Simplifies Integration :**Unit testing may reduce uncertainty in the units themselves and can be used in a bottom-up testing style approach. By testing the parts of a program first and then testing the sum of its parts, integration testing becomes much easier.

**4 ) Documentation :**Unit testing provides a sort of living documentation of the system. Developers looking to learn what functionality is provided by a unit, and how to use it, can look at the unit tests to gain a basic understanding of the unit's interface (API).Unit test cases embody characteristics that are critical to the success of the unit. These characteristics can indicate appropriate/inappropriate use of a unit as well as negative behaviors that are to be trapped by the unit. A unit test case, in and of itself, documents these critical characteristics, although many software development environments do not rely solely upon code to document the product in development.

**5.2 : INTEGRATION TESTING**

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

**5.2.1 Purpose**

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These "design items", i.e., assemblages (or groups of units), are exercised through their interfaces using black-box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test whether all the components within assemblages interact correctly, for example across procedure calls or process activations, and this is done after testing individual modules, i.e., unit testing. The overall idea is a "building block" approach, in which verified assemblages are added to a verified base which is then used to support the integration testing of further assemblages.Software integration testing is performed according to the software development life cycle (SDLC) after module and functional tests. The cross-dependencies for software integration testing are: schedule for integration testing, strategy and selection of the tools used for integration, define the cyclomatical complexity of the software and software architecture, reusability of modules and life-cycle and versioning management.Some different types of integration testing are big-bang, top-down, and bottom-up, mixed (sandwich) and risky-hardest. Other Integration Patterns[2] are: collaboration integration, backbone integration, layer integration, client-server integration, distributed services integration and high-frequency integration.

**5.2.1.1 Big Bang**

In the big-bang approach, most of the developed modules are coupled together to form a complete software system or major part of the system and then used for integration testing. This method is very effective for saving time in the integration testing process. However, if the test cases and their results are not recorded properly, the entire integration process will be more complicated and may prevent the testing team from achieving the goal of integration testing.A type of big-bang integration testing is called "usage model testing" which can be used in both software and hardware integration testing. The basis behind this type of integration testing is to run user-like workloads in integrated user-like environments. In doing the testing in this manner, the environment is proofed, while the individual components are proofed indirectly through their use. Usage Model testing takes an optimistic approach to testing, because it expects to have few problems with the individual components. The strategy relies heavily on the component developers to do the isolated unit testing for their product. The goal of the strategy is to avoid redoing the testing done by the developers, and instead flesh-out problems caused by the interaction of the components in the environment. For integration testing, Usage Model testing can be more efficient and provides better test coverage than traditional focused functional integration testing. To be more efficient and accurate, care must be used in defining the user-like workloads for creating realistic scenarios in exercising the environment. This gives confidence that the integrated environment will work as expected for the target customers.

**5.2.1.2 Top-down And Bottom-up**

Bottom-up testing is an approach to integrated testing where the lowest level components are tested first, then used to facilitate the testing of higher level components. The process is repeated until the component at the top of the hierarchy is tested.All the bottom or low-level modules, procedures or functions are integrated and then tested. After the integration testing of lower level integrated modules, the next level of modules will be formed and can be used for integration testing. This approach is helpful only when all or most of the modules of the same development level are ready. This method also helps to determine the levels of software developed and makes it easier to report testing progress in the form of a percentage.Top-down testing is an approach to integrated testing where the top integrated modules are tested and the branch of the module is tested step by step until the end of the related module.Sandwich testing is an approach to combine top down testing with bottom up testing.

**5.3 : SOFTWARE VERIFICATION AND VALIDATION**

**5.3.1 Introduction**

In software project management, software testing, and software engineering, verification and validation (V&V) is the process of checking that a software system meets specifications and that it fulfills its intended purpose. It may also be referred to as software quality control. It is normally the responsibility of software testers as part of the software development lifecycle. Validation checks that the product design satisfies or fits the intended use (high-level checking), i.e., the software meets the user requirements.This is done through dynamic testing and other forms of review.Verification and validation are not the same thing, although they are often confused. Boehm succinctly expressed the difference between

* Validation : Are we building the right product?
* Verification : Are we building the product right?

According to the Capability Maturity Model (CMMI-SW v1.1)

Software Verification: The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.

Software Validation: The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements.

In other words, software verification is ensuring that the product has been built according to the requirements and design specifications, while software validation ensures that the product meets the user's needs, and that the specifications were correct in the first place. Software verification ensures that "you built it right". Software validation ensures that "you built the right thing". Software validation confirms that the product, as provided, will fulfill its intended use.

From Testing Perspective

* Fault – wrong or missing function in the code.
* Failure – the manifestation of a fault during execution.
* Malfunction – according to its specification the system does not meet its specified functionality

Both verification and validation are related to the concepts of quality and of software quality assurance. By themselves, verification and validation do not guarantee software quality; planning, traceability, configuration management and other aspects of software engineering are required.Within the modeling and simulation (M&S) community, the definitions of verification, validation and accreditation are similar:

* M&S Verification is the process of determining that a ⦁ computer model, simulation, or federation of models and simulations implementations and their associated data accurately represent the developer's conceptual description and specifications.
* M&S Validation is the process of determining the degree to which a model, simulation, or federation of models and simulations, and their associated data are accurate representations of the real world from the perspective of the intended use(s).
* **5.3.2 Classification of Methods**
* In mission-critical software systems, where flawless performance is absolutely necessary, formal methods may be used to ensure the correct operation of a system. However, often for non-mission-critical software systems, formal methods prove to be very costly and an alternative method of software V&V must be sought out. In such cases, syntactic methods are often used.
* **5.3.3 Test Cases**
* A test case is a tool used in the process. Test cases may be prepared for software verification and software validation to determine if the product was built according to the requirements of the user. Other methods, such as reviews, may be used early in the life cycle to provide for software validation.
* **5.4 : Black-Box Testing**
* Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well.
* **5.4.1 Test Procedures**
* Specific knowledge of the application's code/internal structure and programming knowledge in general is not required. The tester is aware of what the software is supposed to do but is not aware of how it does it. For instance, the tester is aware that a particular input returns a certain, invariable output but is not aware of how the software produces the output in the first place.
* **5.4.2 Test Cases**
* Test cases are built around specifications and requirements, i.e., what the application is supposed to do. Test cases are generally derived from external descriptions of the software, including specifications, requirements and design parameters. Although the tests used are primarily functional in nature, non-functional tests may also be used. The test designer selects both valid and invalid inputs and determines the correct output, often with the help of an oracle or a previous result that is known to be good, without any knowledge of the test object's internal structure.
* **5.5 : White-Box Testing**
* White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT). White-box testing can be applied at the unit, integration and system levels of the software testing process. Although traditional testers tended to think of white-box testing as being done at the unit level, it is used for integration and system testing more frequently today. It can test paths within a unit, paths between units during integration, and between subsystems during a system–level test. Though this method of test design can uncover many errors or problems, it has the potential to miss unimplemented parts of the specification or missing requirements.
* **5.5.1 Levels**
* **1 ) Unit testing :**White-box testing is done during unit testing to ensure that the code is working as intended, before any integration happens with previously tested code. White-box testing during unit testing catches any defects early on and aids in any defects that happen later on after the code is integrated with the rest of the application and therefore prevents any type of errors later on.
* **2 ) Integration testing :**White-box testing at this level are written to test the interactions of each interface with each other. The Unit level testing made sure that each code was tested and working accordingly in an isolated environment and integration examines the correctness of the behaviour in an open environment through the use of white-box testing for any interactions of interfaces that are known to the programmer.
* **3 ) Regression testing :**White-box testing during regression testing is the use of recycled white-box test cases at the unit and integration testing levels.

**5.5.2 Procedures**

White-box testing's basic procedures involves the tester having a deep level of understanding of the source code being tested. The programmer must have a deep understanding of the application to know what kinds of test cases to create so that every visible path is exercised for testing. Once the source code is understood then the source code can be analyzed for test cases to be created. These are the three basic steps that white-box testing takes in order to create test cases:

* Input involves different types of requirements, functional specifications, detailed designing of documents, proper source code, security specifications. This is the preparation stage of white-box testing to layout all of the basic information.
* Processing involves performing risk analysis to guide whole testing process, proper test plan, execute test cases and communicate results. This is the phase of building test cases to make sure they thoroughly test the application the given results are recorded accordingly.
* Output involves preparing final report that encompasses all of the above preparations and results.

**5.5.3 Advantages**

White-box testing is one of the two biggest testing methodologies used today. It has several major advantages:

* Side effects of having the knowledge of the source code is beneficial to thorough testing.
* Optimization of code by revealing hidden errors and being able to remove these possible defects.
* Gives the programmer introspection because developers carefully describe any new implementation.
* Provides traceability of tests from the source, allowing future changes to the software to be easily captured in changes to the tests.
* White box testing give clear, engineering-based, rules for when to stop testing.

**5.5.5 Disadvantages**

Although white-box testing has great advantages, it is not perfect and contains some disadvantages:

* White-box testing brings complexity to testing because the tester must have knowledge of the program, including being a programmer. White-box testing requires a programmer with a high level of knowledge due to the complexity of the level of testing that needs to be done.
* On some occasions, it is not realistic to be able to test every single existing condition of the application and some conditions will be untested.
* The tests focus on the software as it exists, and missing functionality may not be discovere

**5.6 : SYSTEM TESTING**

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called assemblages) or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS). System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification(s).

**CHAPTER 6 : ADVANTAGES**

* Efficiency-oriented solutions.
* Diminishing double or over-bookings.
* Manage guest reservations.
* Key pillar to Pharmacy store’s success- Guest Satisfaction.

**CHAPTER 7 : CONCLUSION**

The conclusion of this project is A Pharmacy store management system is a computerized management system. This system keeps the records of hardware assets besides software of this organization. The proposed system will keep a track of Workers, Recidents, Accounts and generation of report regarding the present status. This project has GUI based software that will help in storing, updating and retrieving the information through various user-friendly menu-driven modules.The project “Pharmacy store Management System” is aimed to develop to maintain the day-to-day state of admission/Vacation of Residents, List of Workers , payment details etc. Main objective of this project is to provide solution for Pharmacy store to manage most there work using computerized process. This software application will help admin to handle customers information, items allocation details, payment details, billing information.etc. Detailed explanation about modules and design are provided in project documentation. The existing system is a manually maintained system. All the Pharmacy store records are to be maintained for the details of each customers, Fee details, Items Allocation , Attendance etc. All these details are entered and retrieved manually,because of this there are many disadvantages like Time Consuming ,updating process, inaccuracy of data.For avoiding this we introduced or proposed a new system in proposed system the computerized version of the existing system. provides easy and quick access over the data.

**CHAPTER 8 : BIBLIOGRAPHY**

* https://www.tutorialspoint.com/index.htm
* https://www.javatpoint.com
* https://www.w3schools.com
* https://html.com