

Project Report On E-Commerce (D3 Fashion Hub)



website :- <https://d3-fashion-hub.netlify.app/>

video :-

<https://drive.google.com/file/d/1xCYpS4faD11Lx37g4TLFNUc77pI48MwI/view?usp=drivesdk>

By

Bhargav Patel (20DCS069)

Devan Patel (20DCS070)

Dharm Patel (20DCS071)

Bachelor of Technology (Computer Science)

(2020)

Under the guidance of

Ritiksha Modi , Professor

Parth Goel , HOD



Devang Patel Institute of Advance Technology and Research

CHARUSAT University,Changa,Anand,Gujarat

ACKNOWLEDGEMENT

We express our sincere regard and indebtedness to our project internal guide Ritiksha Modi, for his valuable time, guidance, encouragement, support and cooperation throughout the duration of our project. We would sincerely like to thank IT Department for giving the opportunity to work on enhancing our technical skills while undergoing this project. This project was done under the guidance of Ritiksha Modi, Head of Department. 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.

We would like to thank **Parth Goel**, Head of Department and whole department for their constant support.

Bhargav Patel (20DCS069)

Devan Patel (20DCS070)

Dharm Patel (20DCS071)

ABSTRACT

Electronic Commerce is the process of doing business through computer networks. A person sitting on his chair in front of a computer can access all the facilities of the Internet to buy or sell the products. Unlike traditional commerce that is carried out physically with effort of a person to go & get products, ecommerce has made it easier for human to reduce physical work and to save time. E-Commerce which was started in early 1990's has taken a great leap in the world of computers, but the fact that has hindered the growth of e-commerce is security. Security is the challenge facing e-commerce today & there is still a lot of advancement made in the field of security. The main advantage of e-commerce over traditional commerce is the user can browse online shops, compare prices and order merchandise sitting at home on their PC. For increasing the use of e-commerce in developing countries the B2B e-commerce is implemented for improving access to global markets for firms in developing countries. For a developing country advancement in the field of e-commerce is essential. The research strategy shows the importance of the e-commerce in developing countries for business applications.

The main objective of the E-commerce Portal is to manage the details of Products, Customer, Shipping, Payment, Category. It manages all the information about Products, Sales, Category, Products. The project is totally built at administrative end and thus only the

administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the Products, Customer, Sales, Shipping. It tracks all the details about the Shipping, Payment, Category

INDEX

S.No.	Index	Page No.
Chapter 1	INTRODUCTION	1-6
1.1	Introduction	1
1.2	Aim	1
1.3	Existing System	2
1.4	Proposed System	2
1.5	Feasibility Study	3-4
1.6	Project Work Schedule	5
1.7	Organisation of Report	6
Chapter 2	SOFTWARE REQUIREMENTS SPECIFICATION	7
2.1	Hardware Requirement	7
2.2	Software Requirement	7
Chapter 3	DESIGN & PLANNING	8-14
3.1	Software Development Life Cycle Model	8
3.2	GENERAL OVERVIEW	9
Chapter 4	IMPLEMENTATION DETAILS	15 - 19
4.1	FRONT END	15 - 17
4.2	BACK END	18 - 19
Chapter 5	TESTING AND IMPLEMENTATION	20- 30
5.1	UNIT TESTING	20 - 21
5.2	INTEGRATION TESTING	22 - 23

5.3	SOFTWARE VERIFICATION AND VALIDATION	24 - 26
5.4	Black-Box Testing	27
5.5	White-Box Testing	28 - 29
5.6	SYSTEM TESTING	30
Chapter 6	RESULTS	31 - 33
Chapter 7	ADVANTAGES	34
Chapter 8	CONCLUSION	35
	BIBLIOGRAPHY	36

CHAPTER 1 : INTRODUCTION

1.1 INTRODUCTION

Customer get many benefits via online shopping this helps e-commerce companies to build long-lasting and profitable relationship with their customers. For making strong relationship with these users it is very important to focus on the customer as a whole and making sense of a flood of real-time information that goes well beyond demographics or shopping behavior. There are two entities who will have the access to the system. One is the admin and another one will be the registered user. Admin can add product details, view all the order details and can also view the sales of the products. User need to register with basic registration details to generate a valid username and password. After the user logins, it can view all the products that are recommended on the homepage compiled by the system based on user's information. From the recommended products, the user can even further view its details and then if interested to buy, the system gives add to cart option for purchasing the product. The system even has an AI bot with the help of which the user can get answers to queries like features, warranty, price etc. details of the products. This AI Bot even converts text to speech. After selecting the product, user can do payment for the particular product online. Users can view their order history of their purchased product.

1.2 AIM

The main aim of e-commerce websites development is to sell products to users. The most successful websites are carefully optimized to achieve a high percentage of purchases. To achieve success e-commerce websites need to integrate all of the latest online closing & upsell techniques available which have been proven to increase the chances that a visitor will purchase. There are many

important elements that go into building a successful e-commerce website such as removing friction during the purchasing process, making the checkout smooth and easy, making the website fast and attractive, up selling users on related products, incentivizing buyers, reducing cart abandonment, nurturing past buyers to buy again, remarketing to past visitors who haven't yet purchased, using the proper payment options, having a mobile ready design and many more things which are needed to develop an e-commerce website.

1.3 EXISTING SYSTEM

This existing system of buying goods has several disadvantages. It requires lots of time to travel to the particular shop to buy the goods. It is having lots of manual work. Since everyone is leading busy life now a days, time means a lot to everyone. Also there are expenses for travelling from house to shop. It is less user-friendly. In current system user must go to shop and order products. It is difficult to identify the required product. More over the shop from where we would like to buy something may not be open. Hence we have to adjust our time with the shopkeeper's time or vendor's time. In current e commerce system user have to go shop to view the description of the product. It is unable to generate different kinds of report.

1.4 PROPOSED SYSTEM

The proposed system helps in building a website to buy, sell products or goods online using internet connection. Unlike traditional commerce that is carried out physically with effort of a person to go and get products, eCommerce has made it easier for human to reduce physical work and to save time. The basic concept of the application is to allow the customer to shop virtually using the Internet and allow customers to buy the items and articles of their desire from the store. E-commerce is fast gaining ground as an accepted and used business paradigm.

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 to the feasibility study.

1)Operational Feasibility

2)Technical Feasibility

3)Economical Feasibility

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.

1.6 Giant Chart

Activity	Time Frame					
	31/7/2021 to 7/8/2021	8/8/2021 to 21/8/21	22/8/21 to 2/10/21	3/10/21 to 23/10/21	24/10/21 to 30/10/21	1/11/21 to 13/11/21
Literature Survey And Planning	Red					
	Green					
Feature Finalization		Red				
		Green				
Front End Development			Red			
			Green			
Back End Development				Red		
				Green		
Testing and Fault Detection					Red	
					Green	
Project Report						Red
						Green

1.7 ORGANISATION OF THE REPORT

1.7.1 INTRODUCTION

This section includes the overall view of the project i.e. the basic problem definition and the general overview of the problem which describes the problem in layman terms. It also specifies the software used and the proposed solution strategy.

1.7.2 SOFTWARE REQUIREMENTS SPECIFICATION

This section includes the Software and hardware requirements for the smooth running of the application.

1.7.3 DESIGN & PLANNING

This section consists of the Software Development Life Cycle model. It also contains technical diagrams” like this Data Flow Diagram and the Entity Relationship diagram.

1.7.4 IMPLEMENTATION DETAILS

This section describes the different technologies used for the entire development process of the Front-end as well as the Back-end development of the application.

1.7.5 RESULTS AND DISCUSSION

This section has screenshots of all the implementation i.e. user interface and their description.

1.7.6 SUMMARY AND CONCLUSION

This section has screenshots of all the implementation i.e. user interface and their description.

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	PHP
3	Database	MySQL
4	IDE	Visual Code
5	Browser	Google Chrome

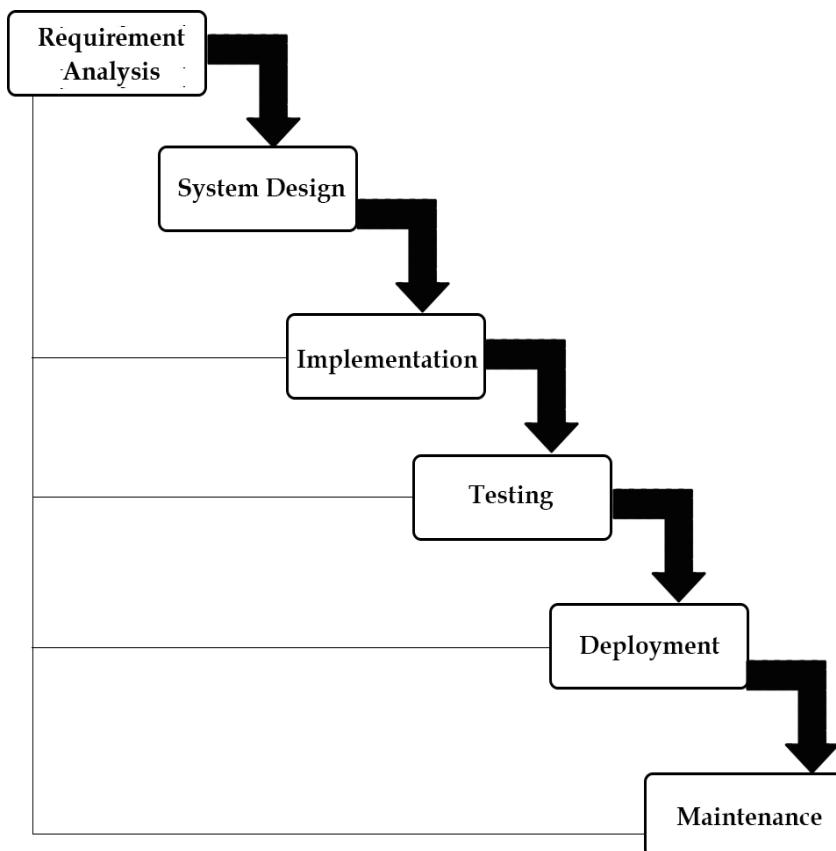
CHAPTER 3 : DESIGN & PLANNING

3.1 Software Development Life Cycle Model

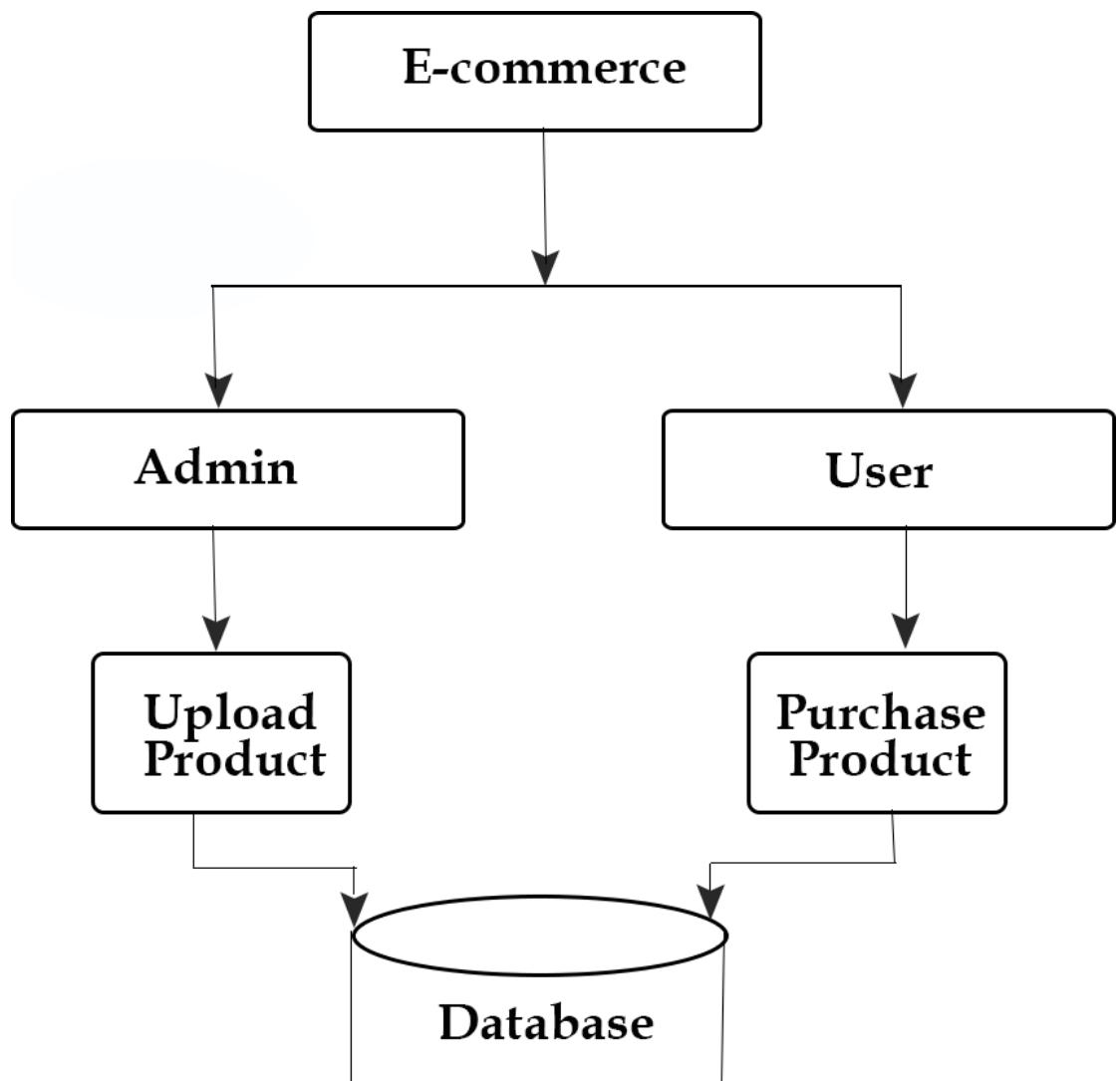
3.1.1 WATERFALL MODEL

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.



3.2 GENERAL OVERVIEW



CHAPTER 4 : IMPLEMENTATION DETAILS

In this Section we will do Analysis of Technologies to use for implementing the project.

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 `` 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 JavaScript



JavaScript is 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.

4.1.3 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.

4.2 : BACK END



4.2.1 PHP

PHP is a server side scripting language that is used to develop Static websites or Dynamic websites or Web applications. PHP stands for Hypertext Pre-processor, that earlier stood for Personal Home Pages. PHP scripts can only be interpreted on a server that has PHP installed. The client computers accessing the PHP scripts require a web browser only. A PHP file contains PHP tags and ends with the extension ".php".

The term PHP is an acronym for PHP: Hypertext Preprocessor. PHP is a server-side scripting language designed specifically for web development. PHP can be easily embedded in HTML files and HTML codes can also be written in a PHP file. The thing that differentiates PHP with client-side language like HTML is, PHP codes are executed on the server whereas HTML codes are directly rendered on the browser.

PHP: Hypertext Preprocessor (or simply PHP) is a general-purpose programming language originally designed for web development. It was originally created by Rasmus Lerdorf in 1994. PHP code may be executed with a command line interface (CLI), embedded into HTML code, or used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in a web server or as a Common Gateway Interface (CGI) executable. The web server outputs the results of the interpreted and executed PHP code, which may be any type of data, 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). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

MySQL is pretty easy to master in comparison with other database software like Oracle Database, or Microsoft SQL Server. MySQL can run on various platforms UNIX, Linux, Windows, etc. You can install it on a server or even in a desktop. Besides, MySQL is reliable, scalable, and fast. The official way to pronounce MySQL is My Ess Que Ell, not My Sequel. However, you can pronounce it whatever you like, who cares?

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.

MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by many popular websites, including Facebook, YouTube, Twitter and so on.

CHAPTER 5 : TESTING AND IMPLEMENTATION

The term implementation has different meanings ranging from the conversion 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 testable 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 Benefits

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.

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 cyclomatic 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.

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.

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.

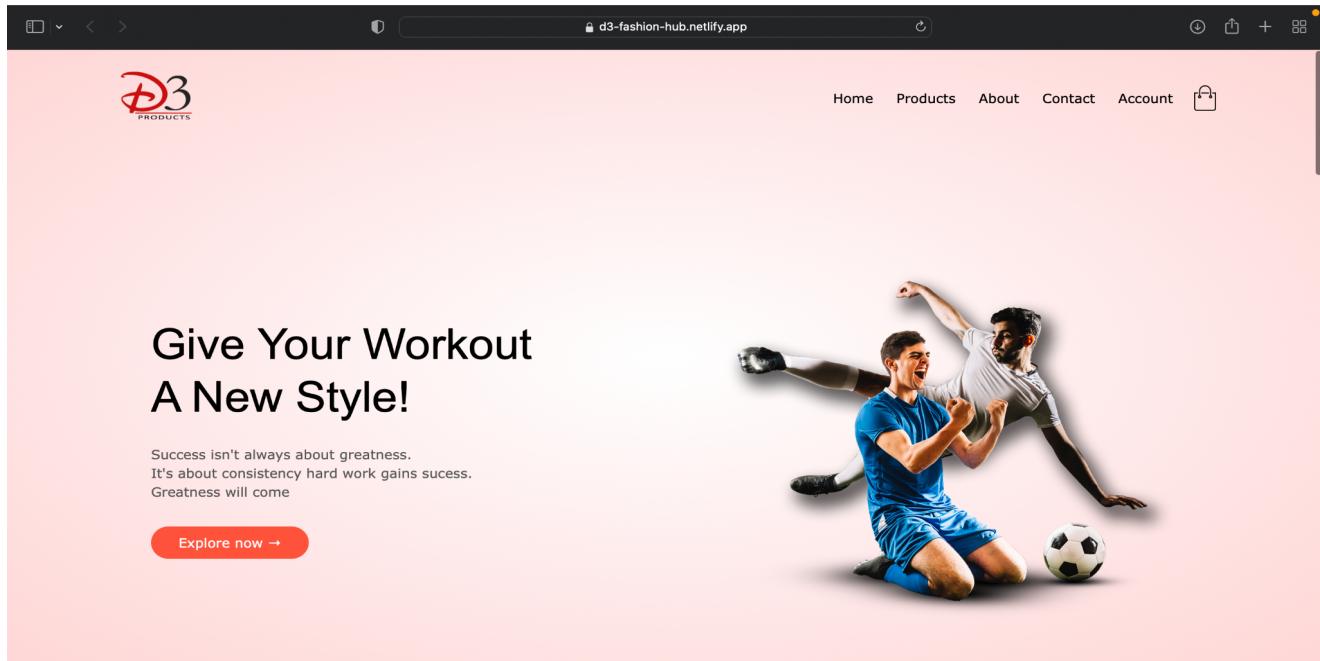
5.6: SYSTEM TESTING

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).

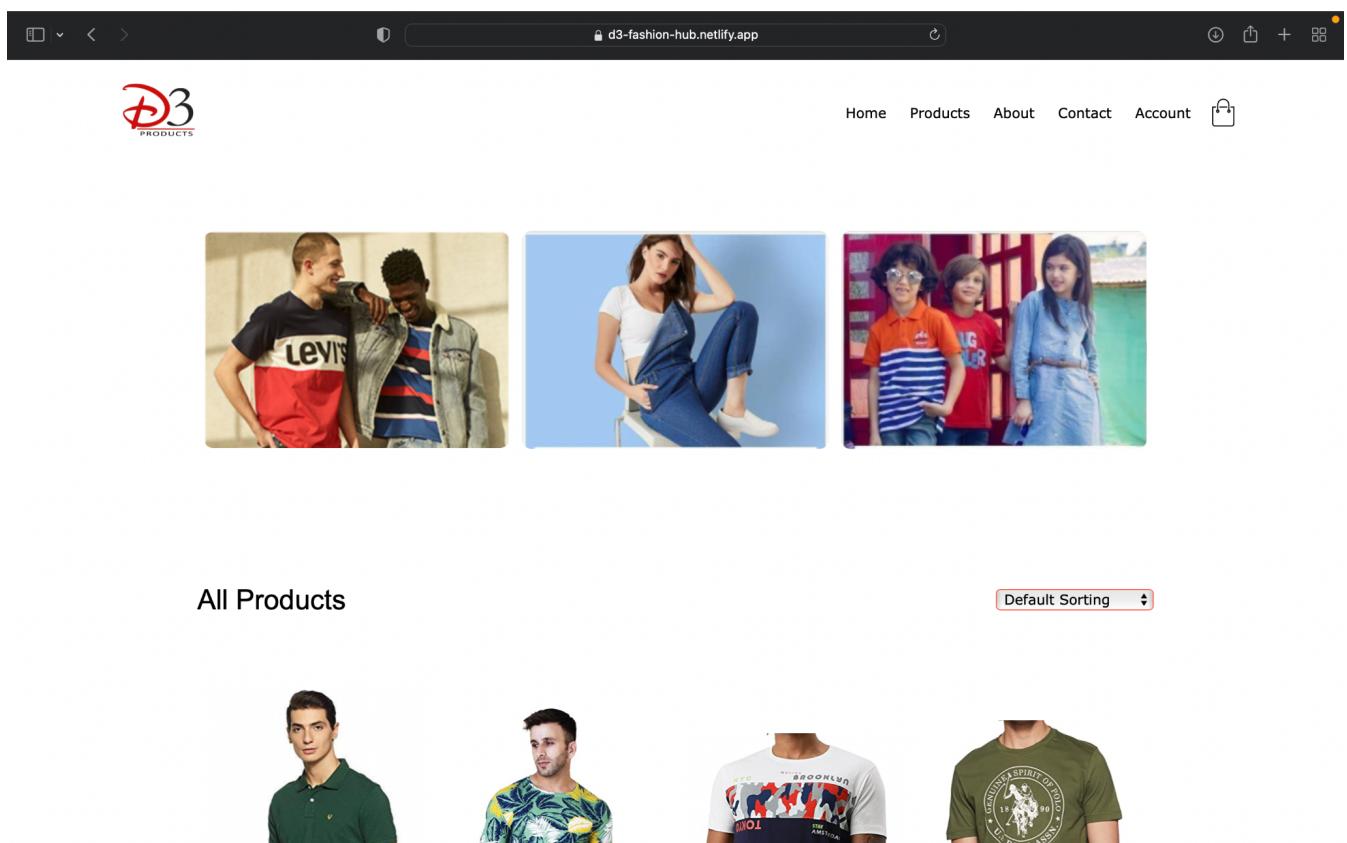
CHAPTER 6 : RESULTS

6.1 HOME PAGE :



The screenshot shows the homepage of d3-fashion-hub.netlify.app. At the top left is the D3 Products logo. The top right features a navigation bar with links for Home, Products, About, Contact, Account, and a shopping cart icon. The main content area has a pink gradient background. On the left, there's a large text block: "Give Your Workout A New Style!" followed by a quote: "Success isn't always about greatness. It's about consistency hard work gains success. Greatness will come." Below the quote is a red button labeled "Explore now →". On the right, there's a dynamic image of two soccer players in action. At the bottom, there are three small thumbnail images showing different product categories.

6.2 Product Page :



The screenshot shows the product page of d3-fashion-hub.netlify.app. The top navigation bar is identical to the home page. The main content area features three large product thumbnails: a man in a Levi's polo shirt, a woman in blue jeans, and a group of four people in casual attire. Below these thumbnails is a section titled "All Products" with a "Default Sorting" dropdown. At the bottom, there are four smaller product thumbnails showing men in various shirts: a green polo, a patterned short-sleeved shirt, a grey t-shirt with a colorful graphic, and a green t-shirt with a horse logo.

6.3 About Us Page :

The screenshot shows a web browser window with the URL `d3-fashion-hub.netlify.app`. The page features a large background image of a man in a clothing store, with the text "About Us" overlaid. Below this, a pink sidebar contains the text "We are the best Fashion Dealer" and a short paragraph of placeholder text. To the right of the sidebar is a photograph of four men in suits.

About Us

We are the best Fashion Dealer

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Fugit consequatur animi odio illum!



6.4 Login Page :

The screenshot shows a web browser window with the URL `d3-fashion-hub.netlify.app`. The page has a pink header and footer. The main content area features a large image of two soccer players in action. To the right is a white box containing a welcome message and login/signup buttons. A red ellipsis button is also present.

Welcome to D3-Fashion-Hub !!!
we hope you are enjoying our
services

• Sign up
• Log in

...

About Us

CHAPTER 7 : ADVANTAGES

Speed up the buying process and save time for customers. Best Quality of services in reasonably low operation cost. Less store setup cost and quick ROI (Return of investments).

No Geographical limitation, tap the global market from the day one.

CHAPTER 8 : CONCLUSION

In general, today's businesses must always strive to create the next best thing that consumers will want because consumers continue to desire their products, services etc. to continuously be better, faster, and cheaper. In this world of new technology, businesses need to accommodate to the new types of consumer needs and trends because it will prove to be vital to their business' success and survival. E-commerce is continuously progressing and is becoming more and more important to businesses as technology continues to advance and is something that should be taken advantage of and implemented. From the inception of the Internet and e-commerce, the possibilities have become endless for both businesses and consumers. Creating more opportunities for profit and advancements for businesses, while creating more options for consumers. However, just like anything else, e-commerce has its disadvantages including consumer uncertainties, but nothing that can not be resolved or avoided by good decision-making and business practices.

CHAPTER 8 : BIBLIOGRAPHY

<https://www.javatpoint.com>

<https://www.w3schools.com>