



**KARNATAK UNIVERSITY’S**

**KARNATAK SCIENCE COLLEGE,**

**DHARWAD**

****

**DEPARTMENT OF COMPUTER SCIENCE**

**BCA AND B.Sc (CS)**

**PROJECT REPORT ON**

**"GOVERNMENT PRINTING PRESS"**

**UNDER THE GUIDANCE OF**

**Shri. Rajshekhar V. B.**

**SUBMITTED BY**

**Priyanka S Ganachari (15M10016)**

**Ruksarbanu M Patil (15M10019)**

**KARNATAK UNIVERSITY'S**

**KARNATAK SCIENCE COLLEGE, DHARWAD**

**Department of Computer Science,**

**B.C.A /B.Sc (C.S) Courses**



2017-2018

CERTIFICATE

This is to certify that **Ms. Priyanka S Ganachari** and **Ms. Ruksarbanu M Patil** has satisfactorily completed the project work entitled **"GOVERNMENT PRINTING PRESS"** for the partial fulfillment of degree in Bachelor of Computer Science from the Karnataka University, Dharwad for the year 2017-2018.

Project Associates Project Guide

Ms. Priyanka S Ganachari (15M10016). Shri. Rajshekhar V.B

Ms. Ruksarbanu M Patil (15M10019).

Examiners Co-ordinator

1.

2.

**ACCEPTANCE LETTER**

Dear Priyanka S G and Ruksarbanu M P, Date:

Mr. Lokesh S A and Mr. Umesh H M. We are the Deputy Director and Assistant Director of Government Printing Press. We are pleased to inform you that we have accepted your Priyanka S Ganachari and Ruksarbanu M Patil students of Karnataka Science College Dharwad (Department of BCA and B.Sc (C.S)) belong proposal for building an application for our firm with the objective of reducing manual work and computerization of the working process. We are most impressed with your comprehensive understanding of the scope of the project and your confidence in being able to complete the project on or before the assigned deadline.

We thank you for submitting your proposal and for satisfying all of the essential requirements in a professional manner.

Yours sincerely,

**Officers Name** **Signature of Officers,**

Mr. Lokesh S A (Deputy Director)

Mr. Umesh H M (Assistant Director)

**ACKNOWLEDGEMENT**

**“Knowledge is a treasure but practice is a key to it”**

This proverb could not be truer than in the context of Project. The successful completion of project is the ecstatic moment in degree student’s life which has a few parallels.

The Successful completion of our project goes half way towards my uniting Efforts and the other invaluable half is attributed to the esteemed help extended to us by innumerable people in one or other way.

We would like to express deep sense of gratitude to our beloved and respected Co-Ordinator **Prof.Shri.Rajshekhar.V.B**. for this incredible co-operation, advice, constant encouragement and timely help throughout the course of my project.

We offer our sincere thanks to Principal **Dr. C. F. Mulimani** without whose assistance the project would not been successful. I am highly privileged in expressing my heartful thanks to **Prof.Shri.Rajshekhar.V.B.** lecturer and assistant professor of Computer Science Department, for the excellent guidance and encouragement given in carrying out this project.

Last but not least, we are grateful to our parents, our friends and all people who have helped us directly or indirectly to make this project successful.

Priyanka S Ganachari (15M10016)

Ruksarbanu M Patil (15M10019)

TABLE OF CONTENTS

1. INTRODUCTION1-2

1.1 PROBLEM DEFINATION1

1.2 PROJECT OVERVIEW 1

1.3 HARDWARE SPECIFICATION 2

1.4 SOFTWARE SPECIFICATION………………………………………….........2

2. LITERATURE SURVEY.3-5

2.1 EXISITING SYSTEM…………………………………………………...3

2.2 PROPOSED SYSTEM…………………………………………………..3

2.3 FEASIBILITY STUDY…………………………………………………..4

3. SYSTEM ANALYSIS AND DESIGN6-19

3.1 REQUIREMENT SPECIFICATION……………………………………6

3.2 ER-DIAGRAM…………………………………………………………..8

3.3 DATA FLOW DIAGRAM……………………………………………...13

4. IMPLEMENTATION 20-52

4.1 LANGUAGE USED……………………………………………………20

4.1.1 PHP………………………………………………………………20

4.1.2 HTML……………………………………………………………22

4.1.3 CSS………………………………………………………………23

4.1.4 BOOTSTRAP……………………………………………………25

4.2 SOFTWARE USED…………………………………………………….27

4.3 CODING………………………………………………………………..36

5. TESTING 53-55

5.1 DETAILS OF TESTING METHODOLOGY…………………………….......53

5.2 SAMPLE TESTING USED ………………………………………………54

6. RESULTS AND SCREENSHOTS 56-60

**7.****CONCUSIONS/ FUTURE SCOPE****…………………………………………………………………..61**

**8. REFERENCES** **……………………………………………………………………………………….62**

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Figure Name** | **Page No** |
| 3.2.1 | E-R Diagram | 12 |
| 3.2.2 | Context Level DFD | 15 |
| 3.2.3 | Level 1 DFD for Admin | 17 |
| 3.2.4 | Level 1 DFD for Govt user | 18 |
| 3.2.5 | Level 1 DFD for Customer | 19 |
| 4.1 | XAMPP Architecture | 20 |
| 6.1 | Screen Shots | 56-60 |

1. **INTRODUCTION**

**1.1 Goal of project:**

The main goal of the application is to maintain the records of purchase, Sales and paper stock details with cash transaction maintenance.

**Objectives of the project:**

This web application has been developed for government printing press located in Sadhankeri, Dharwad. This press has been established to fulfill all the requirements of government offices.

This application allows users from government officesto view various services offered by the printing press such as printing of school text books, providing paper sheets, printing of various kinds of government forms, gazetteer and many other stationary products. It also enables them to place a work order as per their requirements from anywhere with the help of electronic device and internet.

This application also helps the administrator of printing press to maintain stock details, workers details, customer details, employee details and bill details.

In Government printing press all the maintenance procedure takes place manually to overcome this we are providing software to them so that all the data are stored safely in the system.

**1.2 Problem Statement:**

To develop an office automation software for the computer resellers shop which is capable of handling clients details management, product list and billing system.

* 1. **Hardware Requirements**
* Hard disk : 500GB or above.
* Processor : Core i3 or above.
* RAM : 2GB or above.
  1. **Software Requirements**
* Operating Environment : Windows 7 or above.
* Front End : HTML, CSS, Bootstrap.
* Middleware : PHP.
* Backend : MYSQL.
* Scripting : JavaScript.
* Designing Tool : Brackets.
* Server side Software : XAMPP Server.

**2. LITERATURE SURVEY**

Literature survey involves the detail study of existing system and limitations. To overcome the limitations the requirement is reanalyzed and the new system is proposed. The new proposed system should provide more facilities then existing system by residing in same constraints.

**2.1 Existing System**

At present printing press maintain their day transaction manually. Apart from this they can’t maintain paper stock price. As per the records there is no such application which provides the automation of Printing Press. For the past few years companies are trying to enhance existing relations and prove their commitment on a daily basis by taking their time to listen to customers concern and by developing a service focus. At present, the work done is totally manual and hence there is a lot of strain on the person who is entering and updating the data. They generally use billing machine to billing but this machine is not effective for them because it is having lot of drawbacks, apart from this they can’t maintain paper stock price. This project deals with problems on managing and maintaining the different aspects of the company and avoids the problem which carried manually. Identification of drawbacks of the existing system leads to designing of computerized system that will be compatible to the existing system which is more user friendly and GUI oriented. We can overcome the following drawbacks to improve the efficiency of the system.

**2.2 Proposed system**

The proposed system enables the users to view the details of service offered by the printing press from the office itself. To overcome the manual work for the existing system we develop printing press software. This printing press is designed to allow new online store owners a quick and easy means to setup and perform order and other core business over the internet. The overall goals are to find, attract, and win new clients, nurture and retain those the company already has, entice former clients back into the fold, and reduce the costs of marketing and client service. The purpose of this document is to specify requirements and to give guidelines for the development of above said project. In particular it gives guidelines on how to prepare the above said project. This document is intended to be a practical guide for people who developing this software. The proposed system provides scope for the press administrator to maintain all the details efficiently. The application provides the provision for government users to give feedback. Provides one stop solution for storage and retrieval problems

**2.3 Feasibility Study**

A feasibility analysis usually involves a thorough assessment of the operational (need), financial and technical aspects of a proposal. Feasibility study is the test of the system proposal made to identify whether the user needs may be satisfied using the current software and hardware technologies, whether the system will be cost effective from a business point of view and whether it can be developed with the given budgetary constraints. A feasibility study should be relatively cheap and done at the earliest possible time. Depending on the study, the decision is made whether to go ahead with a more detailed analysis.

When a new project is proposed, it normally goes through feasibility assessment. Feasibility study is carried out to determine whether the proposed system is possible to develop with available resources and what should be the cost consideration. Facts considered in the feasibility analysis were

* Technical Feasibility
* Economic Feasibility
* Behavioral Feasibility

**2.3.1 Technical Feasibility**

The technical feasibility in our project it is available to the entire customer and govt. user. Technical feasibility is most important because design of the WebPages and information is available to all. This can be qualified in terms of volumes of data, trends, frequency of updating, cycles of activity etc, in order to give an introduction of technical system. Considering our project it is technically feasible.

**2.3.2 Economic Feasibility**

Comparing cost with existing system and proposal system, there is reduction in the cost of proposal system. In existing system the cost is for raw material is more compare to the proposal system. The technique of cost benefit analysis is often used as a basis for assessing economic feasibility. The proposal system needs some more initial investment than the existing system, but it can be justifiable that it will improve quality of service. The initial investment for our web application is providing computer systems, creating web application, and giving some knowledge about this web application, hosting in internet and get the domain name of this web application and etc. This are the basic initial investment for proposal system.

Thus feasibility study should center along the following points:

* Improvement resulting over the existing method in terms of accuracy, timeliness.
* Cost comparison.
* Estimate on the life expectancy of the hardware.
* It reduces the manual work.

Our project is economically feasible. It does not require much cost to be involved in the overall process. The overall objective is in easing out the recruitment processes.

**2.3.3 Behavioral / Operational Feasibility**

This analysis involves how it will work when it is installed and the assessment of political and managerial environment in which it is implemented. People are inherently resistant to change and computers have been known to facilitate change. The new proposed system is very much useful to the govt. users and customer of government printing press.

1. **SYSTEM REQUIREMENT ANALYSIS AND DESIGN**

**3.1 System Requirement Analysis**

**3.1.1 Definition**

System Analysis is the detailed study of the various operations performed by the system and their relationships within and outside the system. Analysis is the process of breaking something into its parts so that the whole may be understood. System analysis is concerned with becoming aware of the problem, identifying the relevant and most decisional variables, analyzing and synthesizing the various factors and determining an optional or at least a satisfactory solution. During this a problem is identified, alternate system solutions are studied and recommendations are made about committing the resources used to the system.

The Primary step for developing software process is the collecting of the requirements from the admin.We are developing the web application for the -government printing press based on the clients requirement.

In the existing system there is no platform for the government office users to view the details of services offered by the printing press from the office itself. Every time they need to personally visit the government printing press to enquire about the services available & to place the order. The system enables them to view the services that are being provided by the government printing press from the workplace itself. It also enables them to place the office itself.

When the government user place an order the admin of government printing press receives the order request & the government user receives suitable confirmation message based on the availability of the service at government printing press. This saves the time & energy of the government user & improves the efficiency of the process involved. This system enables the government user to give feedback to the admin of government printing press.

Earlier the admin had no platform or facility to advertise the services that are being provided by government printing press. The proposed system enables the admin to being available at the government printing press.

Previously the admin of government printing press had to maintain the details of stock in ledger books which generated unnecessary overhead costs. The proposed system enables the admin to efficiently maintain stock details & provides one stop solution for the storage & retrieval problems.

Similarly the admin of government printing press used to maintain employee, bill & delivery details in ledger books & thus generated overhead costs. Thus the client asked us to solve this problem. The proposed system efficiently solves this problem without generating overhead costs.

**3.1.2 Functional Requirements**

Functional requirements are statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.

**3.1.3 Non-Functional Requirements**

Requirements, which are not related to functional aspect of software, fall into this category. They are implicit or expected characteristics of software, which users make assumption of.

**3.1.4 Performance Requirements**

* Performance: The application takes minimum 2-3 seconds to validate the information or execute the query.
* Reliability: Since the validation is done on each stage the information is processed in a correct and accurate form.
* Safety: The backup of database is always maintained in a local server. & information is securely transferred using https(Hyper-Text Transfer Protocol)

**3.2 Designing**

**3.2.1 ER diagram**

An entity-relationship diagram is a data modeling technique that creates a graphical representation of the entities, and the relationships between entities, within an information system. An entity-relationship model (ERM) is an abstract and conceptual representation of data. Entity-relationship 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. Diagrams created by this process are called entity-relationship diagrams, ER diagrams, or ERDs.

The first stage of information system design uses these models during the requirements analysis to describe information needs or the type of information that is to be stored in a database. The data modeling technique can be used to describe any ontology (i.e. an overview and classifications of used terms and their relationships) for a certain area of interest. In the case of the design of an information system that is based on a database, the conceptual data model is, at a later stage (usually called logical design), mapped to a logical data model, such as the relational model; this in turn is mapped to a physical model during physical design. Sometimes, both of these phases are referred to as "physical design".

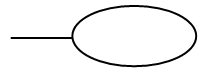
**There are three main components of an E-R Diagram:**

* The entity is a person, object, place or event for which data is collected. For example, if you consider the information system for a business, entities would include not only customers, but the customer’s address, and orders as well. The entity is represented by a rectangle and labeled with a singular noun.
* The relationship is the interaction between the entities. In the example above, the customer places an order, so the word “places” defines the relationship between that instance of a customer and the order or orders that they place. A relationship may be represented by a diamond shape, or more simply, by the line connecting the entities. In either case, verbs are used to label the relationships.
* The cardinality defines the relationship between the entities in terms of numbers. An entity may be optional: for example, a sales representative could have no customers or could have one or many customers; or mandatory: for example, there must be at least one product listed in an order. There are several different types of cardinality notations; crow’s foot notation, used here, is a common one. In crow’s foot notation, a single bar indicates one, a double bar indicates one and only one (for example, a single instance of a product can only be stored in one warehouse), a circle indicates zero, and a crow's foot indicates many. The three main cardinal relationships are: one-to-one, expressed as 1:1; one-to-many, expressed as 1: N; and many-to-many, expressed as M: N.

##### Weak Entity:



A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone. **Attribute:**

****

##### Each entity has attributes or particular properties that describe the entity. Most of the data in a database consists of values of attributes. The set of all possible values of an attribute is the attribute domain. In an ER model, an attribute name appears in an oval that has a line to the corresponding entity box.

##### Key attributes:



A key attribute is the unique, distinguishing characteristic of the entity. An attribute or set of attributes that uniquely identifies a particular entity is a key. A key attribute in an ER Diagram is represented by an oval that has a line inside it and a line to the corresponding entity box. For example, an employee's social security number might be the employee's key attribute.

##### Multi-valued attribute:



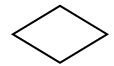
A multi-valued attribute can have more than one value. We indicate this with a double oval. For example, an employee entity can have multiple skill values.

**Derived attribute:**

****

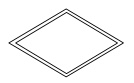
A derived attribute is based on another attribute. It is denoted by a oval and dotted line within it. For example, an employee's monthly salary is based on the employee's annual salary.

##### Relationships:



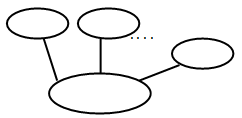
Relationships illustrate how two entities share information in the database structure. An association among entities is called a relationship. An attribute can also be a property of a relationship set. The association among the entities is described as one-to-one, one-to-many, many-to-many. A relationship is indicated by a rhombus.

##### Identifying relationship:



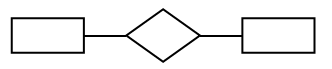
Identifying relationship is denoted by double rhombus.

##### Composite Attribute



A composite attribute has multiple components and each component is atomic or composite. We illustrate this composite nature in the ER model by branching off the component attributes.

##### Total Participation:

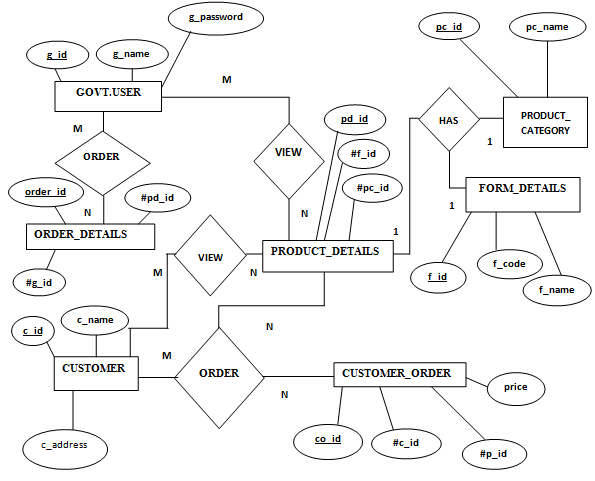


Total participation is represented by a double line.

##### Cardinality:

Cardinality specifies how many instances of an entity relate to one instance of another entity. Ordinarily is also closely linked to cardinality. While cardinality specifies the occurrences of a relationship, ordinarily describes the relationship as either mandatory or optional. In other words, cardinality specifies the maximum number of relationships and ordinarily specifies the absolute minimum number of relationship.

**Entity Relationship Diagram**



**Fig 3.2.1: ER-Diagram**

**3.3 DATA FLOW DIAGRAMS (DFD)**

A data flow diagram (DFD) is a significant modeling technique for analyzing and constructing information processes. Data-flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design).On a DFD, data items flow from an external data source or an internal data store to an Internal data store or an external data sink, via an internal process. A DFD provides no information about the timing or ordering of processes, or about whether processes will operate in sequence or in parallel. It is therefore quite different from a flowchart.

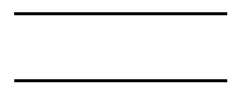
This shows the flow of control through an algorithm, allowing a reader to determine what operations will be performed, in what order, and under what circumstances, but not what kinds of data will be input to and output from the system, nor where the data will come from and go to, nor where the data will be stored (all of which are shown on a DFD). Data-flow diagrams provide the end user with a physical idea of where the data they input ultimately has an effect upon the structure of the whole system from order to dispatch to report. How any system is developed can be determined through a data-flow diagram.

With a data-flow diagram, users are able to visualize how the system will operate, what the system will accomplish, and how the system will be implemented. A designer usually draws a context-level DFD showing the relationship between the entities inside and outside of a system as one single step. This basic DFD can be then disintegrated to a lower level diagram demonstrating smaller steps exhibiting details of the system that is being modeled. Numerous levels may be required to explain a complicated system. The different versions are Context Diagrams (Level 0), Partitioned Diagrams (single process only -- one level), functionally decomposed, leveled sets of Data Flow Diagrams.

### Data Flow Diagrams Symbols

A DFD usually comprises of four components. These four components can be represented by four simple symbols. These symbols can be explained in detail as follows: External entities (source/destination of data) are represented by squares; Processes (input-processing-output) are represented by rectangles with rounded corners; Data Flows (Physical or electronic data) are represented by arrows; and finally, Data Stores (physical or electronic like XML files) are represented by open-ended rectangles.

**Data store**



Or



A data store stores data passively for later access. A data store responds to requests to store and access data. It does not generate any operations. A data store allows values to be accessed in an order different from the order in which they were generated. Input flows indicate information or operations that modify the stored data such as adding or deleting elements or changing values. Output flows indicate information retrieved from the store; this information can be an entire value or a component of a value.

**Data flow**

A data flow moves data between processes or between processes and data stores. As such, it represents a data value at some point within a computation and an intermediate value within a computation if the flow is internal to the diagram. This value is not changed. The names of input and output flows can indicate their roles in the computation or the type of the value they move. Data names are preferably nouns. The name of a typical piece of data, the data aspect, is written alongside the arrow.

**External entity**

Or

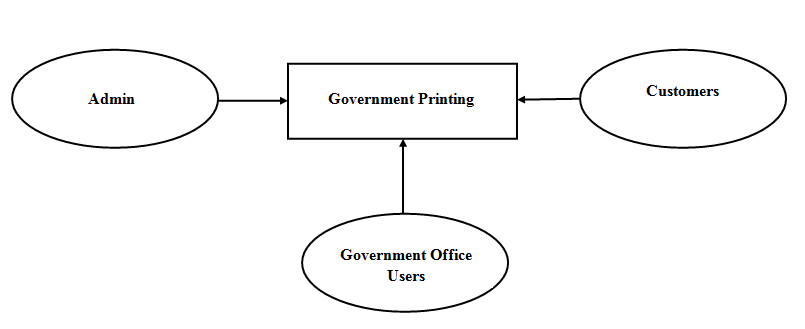
External entities are object outside the system, with which the system communicates. External entities are sources and destination of the system’s inputs and outputs.

**Rules for Creating DFD**

* Processes should be named and numbered for easy reference. Each name should be representative of the process.
* The direction of flow is from top to bottom and from left to right. Data traditionally flow from the source (upper left corner) to the destination although they may flow back to a source. One way to indicate this is to repeat the source symbol as a destination. Since it is used more than once in the DFD, it is marked with a short diagonal in the lower right corner.
* When a process is exploded into the lower-level details, they are numbered.
* The names of data stores, sources and destinations are written in capital letters. Process and data flow names have the first letter of each word capitalized.

**3.2.2 Data Flow Diagrams**

**Context Level DFD**

****

**Fig 3.2.2: Context Level DFD.**

**Description:**

1. **Admin**

* The application should provide the interface for admin to maintain product details, customer details, product category details, form details, govt.user details and employee details efficiently.
* The application should provide the interface for admin to view feedback.

1. **­­­­­­­­­­­Government User**

* The application should provide interface for govt.user and customers to view various kinds of services offered by the government printing press.
* The application should provide interface for govt.user to give feedback.

1. **­­­­­­­­­­­Customer**

* The application also provide interface for the customers to view various kinds of books offered by the government printing press.
* The application should provide interface for customer to give feedback.

**Level 1 DFD**

**Admin:**

**Registration**

**Bill Details**

**Delivery Details**

**Order Details**

**Customer Details**

**Employee** **Details**

**Order Details**

**Feedback Details**

Feedback Details

**Bill Details**

**Delivery Details**

**Admin**

**Login**

**Product Details**

**Product Details** tails

**Employee Details**

**Customer Details**

**Feedback**

**Fig 3.2.3: First Level DFD for Admin.**

**Description:**

* Admin register and login to the Government Printing Press.
* Admin maintains the Customer details, Employee details, Order details, Delivery details, Bill details.
* Admin maintains Product details and Order details of the Government User and Customer.
* Admin view the feedback provided by the Government User only, not for Customer.

**Government Users**

**Govt User**

**Order Details**

**Feedback**

**Product details**

**GOVT USER**

**Registration / Login**

**Confirm Order Request**

**Gives Feedback**

**Product Details**

**Fig 3.2.4: First Level DFD for Govt. user.**

**Description:**

* Government User should register and login to the Government Printing Press.
* Government User can view the forms and stationary product.
* If Government User found the product they needed then they can order product.
* Government User can view only forms and stationary products only,
* Government User sends the feedback to the admin of the Government Printing Press.

**Customer**

**Registration/Login**

**Login**

**Product Details**

**Product Details**

**Order Details**

**CUSTOMER**

**Order Details**

**Feedback**

**Feedback**

**Fig 3.2.5: First Level DFD for Customer.**

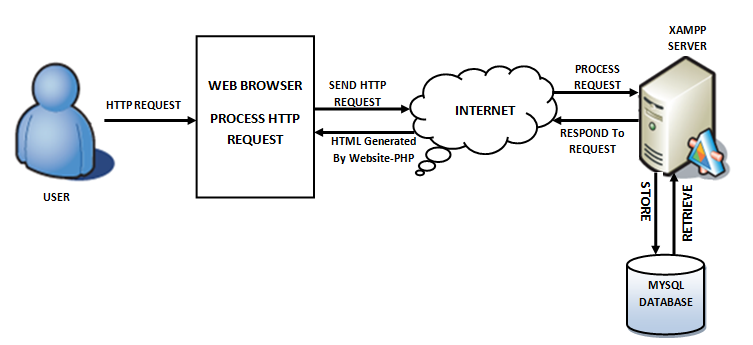
**Description:**

* Customer register and login to the Government Printing Press.
* Customers view the various types of books provided by the Government Printing Press.
* Customers order the book they want.

**4. IMPLEMENTATION**

**4.1 Languages Used**

**PHP Framework**

**Fig 4.1: XAMPP Architecture.**

**4.1.1 Hypertext Pre-Processor (PHP)**

PHP stands for Hypertext Preprocessor created by **Rasmus Lerdorf**, It is a server-side scripting language that powers some of the most popular websites in the world, including Word Press and Face book. However, PHP alone isn't enough in order to build dynamic web sites. To use PHP on a web site, we need a server that can process PHP scripts. XAMPP server allows developers to test PHP scripts locally, this makes it an invaluable piece of your local development environment.

Additionally, dynamic websites are dependent on stored information that can be added and updated easily, this is the main difference between a dynamic application and a static application. However, PHP doesn’t provide a simple, efficient way to store data. This is where a relational database management system like MySQL comes into play.

**Syntax**

<?php

//PHP CODE

?>

PHP originally stands for “Hypertext Pre-Processor” or “Personal Home Page” and was released as a free, open source project. Over time, the language was reworked to meet the needs of its users. In 1997, PHP was renamed to the current “PHP: Hypertext Preprocessor”. PHP is generally used as a server-side scripting language; it is especially well-suited for creating dynamic web pages and client-side GUI applications. PHP generally runs on a web server, taking PHP code as its input and creating web pages as output The scripting language features integrated support for interfacing with databases such as MySQL, which makes it a prime candidate for building all manner of web applications, from simple personal web sites to complex enterprise-level applications.

**Usage**

PHP primarily acts as a filter, taking input from a file or stream containing text and/or PHP instructions and outputs another stream of data; most commonly the output will be HTML. It can automatically detect the language of the user. From PHP 4, the PHP parser compiles input to produce byte code for processing by the Zend Engine, giving improved performance over its interpreter predecessor. Originally designed to create dynamic web pages, PHP’s principal focus is server side scripting, and it is similar to other server-side scripting languages that provide dynamic content from a web server to a client, such as Microsoft’s Active Server Pages, Sun Microsystems’ Java Server Pages, and mod\_perl. PHP has also attracted the development of many frameworks that provide building blocks and a design structure to promote rapid application development (RAD). Some of these include Cake PHP, Symphony, Code Igniter, and Zend Framework, offering features similar to other web application frameworks.

**Speed Optimization**

As with many scripting languages, PHP scripts are normally kept as human-readable source code, even on production web servers. In this case, PHP scripts will be compiled at runtime by the PHP engine, which increases their execution time. PHP scripts are able to be compiled before runtime using PHP compilers as with other programming languages such as C (the language PHP and its extensions are written in). Code optimizers aim to reduce the computational complexity of the compiled code by reducing its size and making other changes that can reduce the execution time with the overall goal of improving performance. The nature of the PHP compiler is such that there are often opportunities for code optimization, and an example of a code optimizer is the Zend Optimizer PHP extension.

Another approach for reducing overhead for high load PHP servers is using PHP accelerators. These can offer significant performance gains by caching the compiled form of a

PHP script in shared memory to avoid the overhead of parsing and compiling the code every time the script runs.

**4.1.2 Hyper Text Markup Language (HTML)**

HTML stands for Hyper Text Markup Language, which is the most widely used language on Web to develop web pages. HTML was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML 4.01, and this version was published in 2012.

**Syntax**

<! DOCTYPE html>

<html>

<head>

<title>This is a Title </title>

</head>

<body>

<p>Hello World!</p>

</body>

</html>

**<! DOCTYPE>**

This tag defines the document type and HTML version.

**<HTML>**

This tag encloses the complete HTML document and mainly comprises of document header which is represented by <head>...</head> and document body which is represented by <body>...</body> tags.

**<HEAD>**

This tag represents the document's header which can keep other HTML tags like <title>, <link> etc.

**<TITLE>**

The <title> tag is used inside the <head> tag to mention the document title.

**<BODY>**

This tag represents the document's body which keeps other HTML tags like <h1>, <div>, <p> etc.

HTML Elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as [interactive forms,](https://en.wikipedia.org/wiki/Fieldset) may be embedded into the rendered page. It provides a means to create structured documents 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/> introduce content into the page directly. Others such as <p>…. </p>, surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

**4.1.3 Cascading Style Sheet (CSS)**

**Cascading Style Sheets** was invented by **Hakon Wium Lie** on October 10, 1994 and maintained through a group of people within the W3C called the CSS Working Group

CSSis the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language. The separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments. There are three types CSS mainly

* External CSS
* Internal CSS
* Inline CSS

**External** style sheets are separate files full of CSS instructions (with the file extension .css). When any web page includes an external style sheet, its look and feel will be controlled by this CSS file. This is how you change a whole website at once. And that's perfect if you want to keep up with the latest fashion in web pages without rewriting every page.

**Internal** styles are placed at the top of each web page document, before any of the content is listed. This is the next best thing to external, because they're easy to find, yet allow you to 'override' an external style sheet -- for that special page that wants to be a nonconformist.

**Inline** styles are placed right where you need them, next to the text or graphic you wish to decorate. You can insert inline styles anywhere in the middle of your HTML code, giving you real freedom to specify each web page element. On the other hand, this can make maintaining web pages a real chore.

**Advantages of CSS**

* **CSS saves time** − You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
* **Pages load faster** − If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
* **Easy maintenance** − To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
* **Superior styles to HTML** − CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
* **Multiple Device Compatibility** − Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
* **Global web standards** − Now HTML attributes are being deprecated and it is being recommended to use CSS. So it’s a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.
* **Offline Browsing** − CSS can store web applications locally with the help of an offline cache. Using of this, we can view offline websites. The cache also ensures faster loading and better overall performance of the website.
* **Platform Independence** − The Script offer consistent platform independence and can support latest browsers as well.

**4.1.4 Bootstrap**

**Bootstrap Definition**

Bootstrap is a powerful mobile first front-end framework for faster and easier web development. It uses HTML, CSS and JavaScript mobile-first website.

**History**

Bootstrap was developed by Mark Otto and Jacob Thornton at Twitter. It was released as an open source.

**Uses of Bootstrap**

* **Easy to use:** Anybody with just basic knowledge of HTML and CSS can start using Bootstrap. Also the Bootstrap official site has a good documentation.
* **Responsive features:** Bootstrap's responsive CSS adjusts to phones, tablets, and desktops. Provides a clean and uniform solution for building an interface for developers. It also provides web based customization. It contains beautiful and functional built-in components which are easy to customize. And best of all it is an open source
* **Mobile-first approach:** In Bootstrap 3, mobile-first styles are part of the core framework
* **Browser compatibility:** Bootstrap is compatible with all modern browsers (Chrome, Firefox, Internet Explorer, Safari, and Opera)
* **Mobile first approach**: Since Bootstrap 3, the framework consists of Mobile first styles throughout the entire library instead of in separate files.
* **Browser Support**: It is supported by all popular browsers.

**Bootstrap Package Includes**

* **CSS**: Bootstrap comes with feature of global CSS settings, fundamental HTML elements styled and enhanced with extensible classes, and an advanced grid system. This is covered in detail in the section **Bootstrap with CSS**.
* **Components:** Bootstrap contains over a dozen reusable components built to provide iconography, dropdowns, navigation, alerts, popovers, and much more. This is covered in detail in the section **Layout Components**.
* **JavaScript Plugins:** Bootstrap contains over a dozen custom jQuery plugins. You can easily include them all, or one by one. This is covered in details in the section **Bootstrap Plugins**
* **Customize:** You can customize Bootstrap's components, LESS variables, and jQuery plugins to get your very own version.

**Responsive Web Design**: Responsive web design is about creating web sites which automatically adjust themselves to look good on all devices, from small phones to large desktops.

## Create First Web Page with Bootstrap

**1. Add the HTML doctype**

Bootstrap uses HTML elements and CSS properties that require the HTML doctype.

Always include the HTML doctype at the beginning of the page, along with the lang attribute and the correct character set:

<! DOCTYPE html>  
<html lang="en">  
 <head>  
     <meta charset="utf-8">   
 </head>  
</html>

**2. Bootstrap 3 is mobile-first**

Bootstrap 3 is designed to be responsive to mobile devices. Mobile-first styles are part of the core framework.

To ensure proper rendering and touch zooming, add the following <meta> tag inside the <head> element:

<meta name="viewport" content="width=device-width, initial-scale=1">

The width=device-width part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The initial-scale=1 part sets the initial zoom level when the page is first loaded by the browser.

**3. Containers**

Bootstrap also requires a containing element to wrap site contents.

There are two container classes to choose from:

The .container class provides a responsive fixed width container

The .container-fluid class provides a full width container, spanning the entire width of the viewport

**4.2 Software Used**

**4.2.1 XAMPP**

**XAMPP** stands for Cross-Platform (X), Apache (A), MySQL (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing purposes. Everything you need to set up a web server – server application (Apache), database (MySQL), and scripting language (PHP) – is included in a simple extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy

**What’s included in XAMPP?**

XAMPP has three primary components. These are:

**1. Apache:** Apache is the actual web server application that processes and delivers web content to a computer. Apache is the most popular web server online, powering nearly 54% of all websites.

**2. MySQL:** Every web application, howsoever simple or complicated, requires a database for storing collected data. MySQL, which is open source, is the world’s most popular database management system. It powers everything from hobbyist websites to professional platforms like WordPress, laravel etc

.

**4.2.2 MYSQL**

SQL is a Relational Database Management System (RDBMS) that runs exclusively under the Windows operating system. One benefit of using Windows exclusively is that we can send and receive E-mail messages based on SQL "events" and we can also let the operating system handle login security. The data base is an organized collection of data. A database management system (DBMS) such as Access, FileMaker Pro, Oracle or a flexible manner. It includes facilities to add, modify or delete data from the SQL Server provides we with the software tools we need to organize that data in database, ask questions (or queries) about the data stored in the database and produce reports summarizing selected contents.

**Database Connection**

<?php

$servername ="localhost";

$username = "username";

$password = "password";

$dbname = "database";

// Create connection

$conn = mysqli\_connect($servername, $username, $password,$dbname);

// Check connection

if (!$conn)

{

die("Connection failed: " . mysqli\_connect\_error());

}

?>

**4.2.2.1 Tables of Government Printing Press in GPP database**

**Admin**

|  |  |
| --- | --- |
| Column | **Type** |
| admin\_id *(Primary)* | int(11) |
| admin\_name | varchar(50) |
| admin\_address | varchar(150) |
| admin\_contact | varchar(50) |
| admin\_email | varchar(50) |
| admin\_designation | varchar(50) |
| admin\_username | varchar(50) |
| admin\_password | varchar(50) |
| admin\_gender | varchar(50) |
| admin\_bankname | varchar(50) |
| admin\_ifsc | varchar(50) |
| admin\_accno | varchar(50) |

**Bill Details**

|  |  |
| --- | --- |
| Column | Type |
| bill\_details\_id *(Primary)* | int(11) |
| govt\_user\_id | int(11) |
| bill\_details\_price | varchar(50) |
| bill\_details\_status | varchar(20) |
| bill\_details\_date | varchar(50) |
| bill\_details\_no | varchar(100) |
| order\_delivery\_status | varchar(100) |

**Customer Bill**

|  |  |
| --- | --- |
| Column | Type |
| customer\_bill\_id *(Primary)* | int(11) |
| customer\_details\_id | varchar(50) |
| customer\_bill\_date | varchar(50) |
| customer\_bill\_status | varchar(20) |
| bill\_order\_no | varchar(100) |

**Delivery Details**

|  |  |
| --- | --- |
| Column | Type |
| delivery\_details\_id *(Primary)* | int(11) |
| employee\_id | int(11) |
| delivery\_date | varchar(50) |
| govt\_user\_id | int(11) |
| labour\_cost | varchar(50) |

**Customer Details**

|  |  |
| --- | --- |
| Column | Type |
| customer\_details\_id *(Primary)* | int(11) |
| customer\_details\_name | varchar(50) |
| customer\_details\_contact | varchar(50) |
| customer\_details\_address | varchar(100) |
| customer\_details\_email | varchar(50) |
| customer\_details\_username | varchar(50) |
| customer\_details\_password | varchar(50) |

**Customer Order**

|  |  |
| --- | --- |
| Column | Type |
| customer\_order\_id *(Primary)* | int(11) |
| product\_details\_id | int(11) |
| customer\_details\_id | int(11) |
| customer\_order\_quantity | varchar(50) |
| customer\_order\_price | varchar(50) |
| customer\_order\_total | varchar(50) |
| customer\_order\_sgst | varchar(50) |
| customer\_order\_cgst | varchar(50) |
| customer\_order\_finalamount | varchar(50) |
| customer\_order\_date | varchar(50) |
| customer\_order\_status | varchar(50) |
| customer\_order\_no | varchar(100) |

**Employee**

|  |  |
| --- | --- |
| Column | Type |
| employee\_id *(Primary)* | int(11) |
| employee\_fullname | varchar(50) |
| employee\_designation | varchar(50) |
| employee\_gender | varchar(50) |
| employee\_age | varchar(50) |
| employee\_contact | varchar(50) |
| employee\_bankname | varchar(50) |
| employee\_ifsc | varchar(50) |
| employee\_accno | varchar(50) |
| employee\_salary | varchar(50) |
| employee\_address | varchar(150) |

**Feedback**

|  |  |
| --- | --- |
| Column | Type |
| feedback\_id *(Primary)* | int(11) |
| feedback\_name | varchar(50) |
| feedback\_subject | varchar(50) |
| feedback\_description | varchar(150) |

**Form Details**

|  |  |
| --- | --- |
| Column | Type |
| form\_details\_id *(Primary)* | int(11) |
| form\_details\_name | varchar(50) |
| form\_details\_code | varchar(50) |

**Govt. User**

|  |  |
| --- | --- |
| Column | Type |
| govt\_user\_id *(Primary)* | int(11) |
| govt\_type | varchar(50) |
| govt\_user\_name | varchar(50) |
| govt\_user\_address | varchar(150) |
| govt\_user\_contact | varchar(50) |
| govt\_user\_email | varchar(50) |
| govt\_user\_designation | varchar(50) |
| govt\_user\_landline | varchar(50) |
| govt\_user\_username | varchar(50) |
| govt\_user\_password | varchar(50) |

**Login**

|  |  |
| --- | --- |
| Column | Type |
| login\_id *(Primary)* | int(11) |
| login\_username | varchar(50) |
| login\_password | varchar(50) |
| login\_type | varchar(50) |

**Product Category**

|  |  |
| --- | --- |
| Column | Type |
| product\_category\_id *(Primary)* | int(11) |
| product\_category\_name | varchar(50) |
| product\_category\_description | varchar(150) |

**Order Details**

|  |  |
| --- | --- |
| Column | Type |
| order\_details\_id *(Primary)* | int(11) |
| product\_details\_id | int(11) |
| govt\_user\_id | int(11) |
| order\_details\_quantity | varchar(50) |
| order\_details\_quality | varchar(50) |
| order\_details\_price | varchar(50) |
| order\_details\_total | varchar(50) |
| order\_details\_sgst | varchar(50) |
| order\_details\_cgst | varchar(50) |
| order\_details\_finalamount | varchar(50) |
| order\_details\_reqdate | varchar(50) |
| order\_details\_date | varchar(50) |
| order\_details\_status | varchar(50) |
| order\_details\_no | varchar(100) |

**Product Details**

|  |  |
| --- | --- |
| Column | Type |
| product\_details\_id *(Primary)* | int(11) |
| product\_category\_id | int(11) |
| form\_details\_id | int(11) |
| product\_details\_name | varchar(50) |
| product\_details\_quantity | varchar(50) |
| product\_details\_description | varchar(150) |
| product\_details\_price | varchar(50) |
| product\_details\_image | varchar(200) |

**Features of SQL**

* It is simple English like language and uses simple commands such as SELECT, CREATE, DROP etc.
* It is not having condition loops, variables and most of the commands are single line commands.
* To implement application logics, SQL has got extension language popularly called as PL/SQL (Procedural language of sql).
* The entire SQL has been divided into 4 major categories.
  1. Data Manipulation Language.
  2. Data Definition Language.
  3. Transaction Control language.
  4. Data Control Language.

**Security**

View are basically used as a part of security, means in many organizations ,the end user will never be given original tables & all data entry will be done with the help of views only. But the database administrator will be able to see everything because all the operations done by the different users will come to the same table.

**4.2.3 Brackets (**[open-source](https://en.wikipedia.org/wiki/Open-source) [editor](https://en.wikipedia.org/wiki/HTML_editor))

**Brackets** is an lightweight & [open-source](https://en.wikipedia.org/wiki/Open-source) [editor](https://en.wikipedia.org/wiki/HTML_editor) written in [HTML](https://en.wikipedia.org/wiki/HTML), [CSS](https://en.wikipedia.org/wiki/CSS), and [JavaScript](https://en.wikipedia.org/wiki/JavaScript) with a primary focus on [web development](https://en.wikipedia.org/wiki/Web_Development). It was created by [Adobe Systems](https://en.wikipedia.org/wiki/Adobe_Systems), licensed under the [MIT License](https://en.wikipedia.org/wiki/MIT_License), and is currently maintained on [GitHub](https://en.wikipedia.org/wiki/GitHub). Brackets are available for Mac OS X, Windows and Linux.

Brackets provide several featuresincluding:

* Quick Edit
* Live Preview

**Quick Edit**

Quick edit enables inline editing of CSS, Color Property, and JavaScript elements for developers. This built-in feature can be applied to multiple functions or properties simultaneously and all updates are applied directly to the file associated with the changed elements.

* [**HTML**](https://en.wikipedia.org/wiki/HTML)**File**
* Applying quick edit to [HTML elements](https://en.wikipedia.org/wiki/HTML_element) will display all corresponding [CSS](https://en.wikipedia.org/wiki/CSS) properties in a box beneath the selected element. Users can choose to create new [CSS](https://en.wikipedia.org/wiki/CSS) rules directly within the [editor](https://en.wikipedia.org/wiki/HTML_editor) and edit a [tag's](https://en.wikipedia.org/wiki/HTML_element) [CSS](https://en.wikipedia.org/wiki/CSS) properties inline without leaving the context of the [HTML](https://en.wikipedia.org/wiki/HTML) file.
* [**JavaScript**](https://en.wikipedia.org/wiki/JavaScript)**File**
* On [JavaScript](https://en.wikipedia.org/wiki/JavaScript) functions, quick edit performs the same procedure as with [HTML elements](https://en.wikipedia.org/wiki/HTML_element) but displays the selected function’s body within the drop down box. All updates to the function’s body will propagate and update directly within the corresponding [JavaScript](https://en.wikipedia.org/wiki/JavaScript) file.
* **Files containing**[**Hex**](https://en.wikipedia.org/wiki/Hexadecimal)**or**[**RGB**](https://en.wikipedia.org/wiki/RGB)**color properties**
* For color properties, quick edit will return an inline color picker for previewing and color adjustment functionality.

**Live Preview**

When one clicks the respective code snippet in [CSS](https://en.wikipedia.org/wiki/CSS)/[HTML](https://en.wikipedia.org/wiki/HTML) the web browser immediately shows the output relating to that code snippet in web browser. This feature is termed as Live Highlight. Also, the feature Live Preview in Brackets pushes code edits instantly to the [browser](https://en.wikipedia.org/wiki/Web_browser) to present an updated webpage as the developers modify the code. Brackets contains a [Node.js](https://en.wikipedia.org/wiki/Node.js) backend that predicts what the code does as the developer types the code

**Major Features**

* **Live Preview:** It launches a new browser window showing the current page that not only does not require a manual refresh when you change a file, but also as you type. As such, Brackets feels like an extension of your browser’s Developer tools. This feature is incredibly useful, as it can combine the advantages of coding in the browser with those of an editor.
* **Quick Edit:** When you are editing HTML code, if you click a tag with a corresponding CSS declaration situated in a linked file and hit Ctrl/Command+E, an inline editor appears allowing you to quickly edit that rule.
* **JSLint:** It works with your JavaScript code upon saving, with its results displayed as a panel below the main editing window.
* Brackets implements code completion especially meant for web designers and front-end development.
* Additional features such as command line integration, multiple code selection, and quick open and quick find, add to the highlights of Brackets code editor.

**Brackets Extensions**

**1. Beautify**

[Beautify](https://github.com/brackets-beautify/brackets-beautify) Brackets extension makes your code look good. Not only does it make your files more readable, but it also gives you a handy shortcut key to beautify those files — CMD-Shift-L (Mac) and Ctrl-Shift-L (Win). JavaScript code can be consistently formatted and made more readable using Beautify.

**2. Autoprefixer**

[Autoprefixer](https://github.com/mikaeljorhult/brackets-autoprefixer) is a postprocessor for dealing with vendor prefixes in the best possible way. Autoprefixer parses CSS files and adds vendor prefixes to CSS rules. All you have to do is to add it to your asset building tool. Autoprefixer cleans outdated prefixes as well.

**3. Emmet**

[Emmet](https://github.com/emmetio/brackets-emmet) helps improve your HTML and CSS workflow by letting you type in CSS-like expressions that can be dynamically parsed and produce an output according to what you actually typed in the abbreviation.

Emmet abbreviations look like CSS selectors but are parsed in runtime and turned into a structured code block with just a single key press.

Once you learn the abbreviations, it’s a helpful way to speed up your workflow.

**4. Minifier**

[Minifier](https://github.com/alfredxing/brackets-minifier/) minifies JavaScript and CSS files in brackets and saves them as filename.min.ext

To minify a file, use the keyboard shortcut CMD+M or Ctrl+M. It compresses files and saves your time as it minifies the relevant JavaScript automatically so you can continue to work and have the file minified and fully ready for deployment.

**5. HTML Skeleton**

[HTML Skeleton](https://github.com/le717/brackets-html-skeleton) is a collection of a set of tags required by every HTML webpage you build. The tags that make up the skeleton tell browsers what kind of file it is reading and without the skeleton, HTML files will not be rendered correctly in web browsers. This is a must-have extension if you regularly use HTML to build pages in Brackets.

**6. HTML Wrapper**

[HTML Wrapper](https://github.com/rcaferati/brackets-html-wrapper) is a Brackets extension that formats nav and selects tags with a single command. You can use it by selecting a list of items encapsulated by the desired tag, and then run the command.

**7. JavaScript and CSS CDN Suggestions**

[Brackets CDN extension](https://github.com/szdc/brackets-cdn-suggestions) lets you work with CDN services. As you already know, a Content Delivery Network (CDN) saves your web server resources by loading files via cloud servers.

**4.3 Coding**

**Login page**

<!DOCTYPE html>

<html>

<head lang="en">

<? Php require ('1\_metatags.php');?>

</head>

<body>

<header id="header" class="">

<?php require('2\_header.php'); ?>

</header>

<nav class="navbar">

<? php require('3\_navbar.php'); ?>

</nav>

<div id="content" class="site-content">

<div class="container">

<div class="row">

<div class="col-md-8 col-sm-7 col-xs-12 main-content">

<div class="sv-contact-page sv-login-page">

<form class="sv-contact-form" action="logcheck.php" method="post">

<fieldset>

<div>

<label>Username</label>

<input type="text" name="login\_username" required>

</div>

<div>

<label>Password</label>

<input type="password" name="login\_password" required>

</div>

<div>

<button type="submit" class="sv-button btn btn-lg">login</button>

</div>

<div>

<a href="#" class="sv-lost-password">forgot password ?</a>

</div>

</fieldset>

</form>

</div>

</div>

<div class="col-md-4 col-sm-5 col-xs-12">

<aside class="sidebar">

<div class="widget widget-menu">

<h4 class="widget-title">PRODUCT CATEGORIES</h4>

<ul>

<li>

<a href="stationary.php">

<b>STATIONARY</b>

</a>

</li>

<li>

<a href="forms.php">

<b>FORMS</b>

</a>

</li>

<li>

<a href="books.php">

<b>GAZETTEER BOOKS</b>

</a>

</li>

</ul>

</div>

</aside>

</div>

</div>

</div>

</div>

<footer id="footer">

<div class="footer-top widget-category">

<div class="container">

<div class="row">

<div class="col-md-3 col-sm-6 col-xs-12">

<h3 class="sv-footer-header">quick links</h3>

<ul>

<li>

<a href="index.php">

Home

</a>

</li>

<li>

<a href="aboutus.php">

About Us

</a>

</li>

<li>

<a href="stationary.php">

Stationary

</a>

</li>

<li>

<a href="forms.php">

Forms

</a>

</li>

<li>

<a href="books.php">

Books

</a>

</li>

<li>

<a href="feed-back.php">

Feedback

</a>

</li>

<li>

<a href="contact.php">

Contact us

</a>

</li>

</ul>

</div>

<div class="col-md-3 col-sm-6 col-xs-12">

<?php

require('db\_connect.php');

require('valid.php');

?>

<form class="sv-contact-form" name="feedback\_form" id="feedback\_form" action="feedback\_insert.php" method="post" enctype="multipart/form-data">

<center><h4 style="color:white" align="">FEEDBACK</h4></center>

<table border="0" align="center">

<tr>

<td><label style="color:white">Name &nbsp;</label></td>

<td style="padding:2px;"><input type="text" name="feedback\_name" id="feedback\_name" class="form-control" required ></td>

<br>

</tr>

<tr>

<td><label style="color:white">Subject &nbsp;</label></td>

<td style="padding: 2px ;">< input type="text" name="feedback\_subject" id="feedback\_subject" class="form-control" required></td>

</tr>

<tr>

<td><label style="color:white">Description &nbsp;</label></td>

<td style="padding:2px;"><textarea name="feedback\_description" id="feedback\_description" class="form-control" required> </textarea></td>

</tr>

<tr>

<td style="padding:2px;" colspan="2" align="center"><button type="submit" class="btn btn-primary">Submit</button></td>

</tr>

</table>

</form>

</div>

</div>

</div>

</div>

<div class="footer-bottom">

<div class="container">

<div class="row">

<div class="col-md-6 vcenter col-sm-6 col-xs-12 sv-copyright">

<p>&copy; 2018 - GOVERNAMENT PRINTING PRESS <a href="#" target="\_blank">Developed By Ruksaar And Priyanka</a></p>

</div>

</div>

</div>

</div>

</footer>

<script src="js/jquery-3.2.1.min.js"></script>

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

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

<script src="js/jquery.mCustomScrollbar.concat.min.js"></script>

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

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

<?php include('valid.php');?>

</body>

</html>

**PRODUCT FORM PAGE**

<!DOCTYPE html>

<html lang="en">

<head>

<?php include('meta.php');?>

</head>

<body class="fix-sidebar">

<div class="preloader">

<div class="cssload-speeding-wheel"></div>

</div>

<div id="wrapper">

<?php include('nav.php');?>

<!-- End Top Navigation -->

<?php include('leftnewbar.php');?>

<!-- Page Content -->

<div id="page-wrapper">

<div class="container-fluid">

<div class="row bg-title">

<div class="col-lg-3 col-md-4 col-sm-4 col-xs-12">

<h4 class="page-title">Admin</h4>

</div>

</div>

<div class="row">

<div class="col-sm-12">

<div class="white-box">

<h3 class="box-title m-b-0"><b>Enter Product Details</b></h3><br>

<form name="product\_details\_form" id="product\_details\_form" action="product\_details\_insert.php" method="post" enctype="multipart/form-data" class="form-horizontal" novalidate>

<div class="form-group item">

<label class="col-md-12">Product\_category\_id</label>

<div class="col-md-12">

<?php

require('db\_connect.php');

$sql="SELECT \* FROM product\_category";

$res=mysqli\_query($conn,$sql);

?>

<select name="product\_category\_id" id="product\_category\_id" class="form-control" required>

<option value="">--Select Value--</option>

<?php

while ($row = mysqli\_fetch\_array($res,MYSQLI\_ASSOC))

{ ?>

<option value="<?php echo $row['product\_category\_id']; ?>"><?php echo $row['product\_category\_name']; ?></option>

<?php

} ?>

</select>

</div>

</div>

<div class="form-group item">

<label class="col-md-12">Form\_details\_id</label>

<div class="col-md-12">

<?php

require('db\_connect.php');

$sql1="SELECT \* FROM form\_details";

$res1=mysqli\_query($conn,$sql1);

?>

<select name="form\_details\_id" id="form\_details\_id" class="form-control">

<option value="">--Select Value--</option>

<?php

while ($row1 = mysqli\_fetch\_array($res1,MYSQLI\_ASSOC))

{

?>

<option value="<?php echo $row1['form\_details\_id']; ?>"><?php echo $row1['form\_details\_name']; ?></option>

<?php

}

?>

</select>

</div>

</div>

<div class="form-group item">

<label class="col-md-12">NAME</label>

<div class="col-md-12">

<input type="text" class="form-control" pattern="alpha" placeholder="product\_details\_name" name="product\_details\_name" id="product\_details\_name" required="required" />

</div>

</div>

<div class="form-group item">

<label class="col-md-12">QUANTITY</label>

<div class="col-md-12">

<input type="text" class="form-control" pattern="numeric" placeholder="product\_details\_quantity" name="product\_details\_quantity" id="product\_details\_quantity" required="required" />

</div>

</div>

<div class="form-group item">

<label class="col-md-12">DESCRIPTION</label>

<div class="col-md-12">

<input type="text" class="form-control" pattern="alpha" placeholder="product\_details\_description" name="product\_details\_description" id="product\_details\_description" required="required" />

</div>

</div>

<div class="form-group item">

<label class="col-md-12">PRICE</label>

<div class="col-md-12">

<input type="text" class="form-control" pattern="numeric" placeholder="product\_details\_price" name="product\_details\_price" id="product\_details\_price" required="required" />

</div>

</div>

<div class="form-group item">

<label class="col-md-12">IMAGE</label>

<div class="col-md-12">

<input type="file" class="form-control" placeholder="product\_details\_image" name="product\_details\_image" id="product\_details\_image" required="required" />

</div>

</div>

<button type="submit" class="btn btn-primary">Save</button>

<input type="reset" name="reset" class="btn btn-primary" id="reset" value="Reset" />

<input type="button" name="button" class="btn btn-primary" id="button" value="Cancel" onclick="window.location.href='product\_details\_view.php'" />

</form>

</div>

</div>

</div>

</div>

<footer class="footer text-center"> 2017 &copy; Elite Admin brought to you by themedesigner.in </footer>

</div>

</div>

<?php include('footer.php');?>

<?php include('valid.php');?>

</body>

</html>

**PRODUCT INSERT**

<?php

require("db\_connect.php");

$product\_details\_image=$\_FILES["product\_details\_image"]["name"];

$product\_category\_id=$\_POST['product\_category\_id'];

$form\_details\_id=$\_POST['form\_details\_id'];

$product\_details\_name=$\_POST['product\_details\_name'];

$product\_details\_quantity=$\_POST['product\_details\_quantity'];

$product\_details\_description=$\_POST['product\_details\_description'];

$product\_details\_price=$\_POST['product\_details\_price'];

if($product\_category\_id=="" && $form\_details\_id=="")

{

echo '<script type="text/javascript">

alert("Please select any one of the option");

window.location="product\_details\_form.php";

</script>';

}

else

{

$sql = "INSERT INTO `product\_details` (`product\_category\_id`,`form\_details\_id`,`product\_details\_name`,`product\_details\_quantity`,`product\_details\_description`,`product\_details\_price`,`product\_details\_image`) VALUES ('$product\_category\_id','$form\_details\_id','$product\_details\_name','$product\_details\_quantity','$product\_details\_description','$product\_details\_price','$product\_details\_image')";

$result = mysqli\_query($conn,$sql);

if(! $result)

{

//die('Could not enter data: ' .mysqli\_error());

echo '<script type="text/javascript">

alert("Record Not Added");

window.location="product\_details\_view.php";

</script>';

}

else

{

echo '<script type="text/javascript">

alert("Record Added Successfully");

window.location="product\_details\_view.php";

</script>';

}

}

?>

**PRODUCT VIEW**

<!DOCTYPE html>

<html lang="en">

<?php require('meta.php');?> <!-- meta data included-->

<body class="fix-sidebar">

<!-- Preloader -->

<div class="preloader">

<div class="cssload-speeding-wheel"></div>

</div>

<div id="wrapper">

<?php require('nav.php'); ?> <!--navigation included-->

<?php require('leftnewbar.php'); ?> <!--left new bar included-->

<div id="page-wrapper">

<div class="container-fluid">

<div class="row bg-title">

<div class="col-lg-3 col-md-4 col-sm-4 col-xs-12">

<h4 class="page-title">Product Details</h4>

</div>

</div>

<div class="row">

<div class="col-sm-12">

<div class="white-box">

<button type="button" onclick="window.location.href='product\_details\_form.php'" class="btn btn-primary">New</button><br><br>

<!--<h3 class="box-title m-b-0">Admin Details</h3>-->

<!--<p class="text-muted m-b-30">Data table example</p>-->

<div class="table-responsive">

<?php

require ("db\_connect.php");

$sql = 'SELECT `product\_details\_id`,`product\_category\_id`,`form\_details\_id`,`product\_details\_name`,`product\_details\_quantity`,`product\_details\_description`,`product\_details\_price`,`product\_details\_image` from `product\_details`';

$result = mysqli\_query($conn,$sql);

if(! $result)

{

echo '<script type="text/javascript">

alert("Record Not Found");

window.location="index.php";

</script>';

}

?>

<table id="myTable" class="table table-striped">

<thead>

<tr>

<td><b>SLNO</b></td>

<td><b>product\_category\_id</b></td>

<td><b>form\_details\_id</b></td>

<td><b>product\_details\_name</b></td>

<td><b>product\_details\_quantity</b></td>

<td><b>product\_details\_description</b></td>

<td><b>product\_details\_price</b></td>

<td><b>product\_details\_image</b></td>

<td><b>Edit</b></td>

<td><b>Delete</b></td>

</tr>

</thead>

<tbody>

<?php

$sl=1;

while ($row = mysqli\_fetch\_array($result,MYSQLI\_ASSOC))

{

?>

<tr>

<td> <?php echo $sl++; ?></td>

<td> <?php echo $row['product\_category\_id']; ?></td>

<td> <?php echo $row['form\_details\_id']; ?></td>

<td> <?php echo $row['product\_details\_name']; ?></td>

<td> <?php echo $row['product\_details\_quantity']; ?></td>

<td> <?php echo $row['product\_details\_description']; ?></td>

<td> <?php echo $row['product\_details\_price']; ?></td>

<td> <img width="50" height="50" src="photos/<?php echo $row['product\_details\_image']; ?>"></td>

<td><a href='product\_details\_edit\_form.php?id=<?php echo $row['product\_details\_id'];?>' onClick="return confirm('Are You Sure Want To Edit ?')" >Edit</a></td>

<td><a href='product\_details\_delete.php?id=<?php echo $row['product\_details\_id'];?>' onClick="return confirm('Do You Really Want To Delete ?')">Delete</a></td>

</tr>

<?php

}

?>

</tbody>

</table>

</div>

</div>

</div>

</div>

<div class="right-sidebar">

<div class="slimscrollright">

<div class="rpanel-title"> Service Panel <span><i class="ti-close right-side-toggle"></i></span> </div>

<div class="r-panel-body">

<ul>

<li><b>Layout Options</b></li>

<li>

<div class="checkbox checkbox-info">

<input id="checkbox1" type="checkbox" class="fxhdr">

<label for="checkbox1"> Fix Header </label>

</div>

</li>

<li>

<div class="checkbox checkbox-warning">

<input id="checkbox2" type="checkbox" checked="" class="fxsdr">

<label for="checkbox2"> Fix Sidebar </label>

</div>

</li>

<li>

<div class="checkbox checkbox-success">

<input id="checkbox4" type="checkbox" class="open-close">

<label for="checkbox4"> Toggle Sidebar </label>

</div>

</li>

</ul>

<ul id="themecolors" class="m-t-20">

<li><b>With Light sidebar</b></li>

<li><a href="javascript:void(0)" theme="default" class="default-theme">1</a></li>

<li><a href="javascript:void(0)" theme="green" class="green-theme">2</a></li>

<li><a href="javascript:void(0)" theme="gray" class="yellow-theme">3</a></li>

<li><a href="javascript:void(0)" theme="blue" class="blue-theme working">4</a></li>

<li><a href="javascript:void(0)" theme="purple" class="purple-theme">5</a></li>

<li><a href="javascript:void(0)" theme="megna" class="megna-theme">6</a></li>

<li><b>With Dark sidebar</b></li>

<br/>

<li><a href="javascript:void(0)" theme="default-dark" class="default-dark-theme">7</a></li>

<li><a href="javascript:void(0)" theme="green-dark" class="green-dark-theme">8</a></li>

<li><a href="javascript:void(0)" theme="gray-dark" class="yellow-dark-theme">9</a></li>

<li><a href="javascript:void(0)" theme="blue-dark" class="blue-dark-theme">10</a></li>

<li><a href="javascript:void(0)" theme="purple-dark" class="purple-dark-theme">11</a></li>

<li><a href="javascript:void(0)" theme="megna-dark" class="megna-dark-theme">12</a></li>

</ul>

<ul class="m-t-20 chatonline">

<li><b>Chat option</b></li>

<li>

<a href="javascript:void(0)"><img src="plugins/images/users/varun.jpg" alt="user-img" class="img-circle"> <span>Varun Dhavan <small class="text-success">online</small></span></a>

</li>

<li><a href="javascript:void(0)"><img src="plugins/images/users/genu.jpg" alt="user-img" class="img-circle"> <span>Genelia Deshmukh <small class="text-warning">Away</small></span></a>

</li>

<li>

<a href="javascript:void(0)"><img src="plugins/images/users/ritesh.jpg" alt="user-img" class="img-circle"> <span>Ritesh Deshmukh <small class="text-danger">Busy</small></span></a>

</li>

<li><a href="javascript:void(0)"><img src="plugins/images/users/arijit.jpg" alt="user-img" class="img-circle"> <span>Arijit Sinh <small class="text-muted">Offline</small></span></a>

</li>

<li>

<a href="javascript:void(0)"><img src="plugins/images/users/govinda.jpg" alt="user-img" class="img-circle"> <span>Govinda Star <small class="text-success">online</small></span></a>

</li>

<li>

<a href="javascript:void(0)"><img src="plugins/images/users/hritik.jpg" alt="user-img" class="img-circle"> <span>John Abraham<small class="text-success">online</small></span></a>

</li>

<li>

<a href="javascript:void(0)"><img src="plugins/images/users/john.jpg" alt="user-img" class="img-circle"> <span>Hritik Roshan<small class="text-success">online</small></span></a>

</li>

<li>

<a href="javascript:void(0)"><img src="plugins/images/users/pawandeep.jpg" alt="user-img" class="img-circle"> <span>Pwandeep rajan <small class="text-success">online</small></span></a>

</li>

</ul>

</div>

</div>

</div>

</div>

<footer class="footer text-center"> 2017 &copy; Elite Admin brought to you by themedesigner.in </footer>

</div>

</div>

<?php require('footer.php');?>

</body>

</html>

**PRODUCT EDIT PAGE**

<!DOCTYPE html>

<html lang="en">

<head>

<?php include('meta.php');?>

</head>

<body class="fix-sidebar">

<div class="preloader">

<div class="cssload-speeding-wheel"></div>

</div>

<div id="wrapper">

<?php include('nav.php');?>

<?php include('leftnewbar.php');?>

<div id="page-wrapper">

<div class="container-fluid">

<div class="row bg-title">

<div class="col-lg-3 col-md-4 col-sm-4 col-xs-12">

<h4 class="page-title">Admin</h4>

</div>

</div>

<div class="row">

<div class="col-sm-12">

<div class="white-box">

<h3 class="box-title m-b-0"><b>Edit Product Details</b></h3><br>

<?php

require("db\_connect.php");

$product\_details\_id=$\_REQUEST['id'];

$sqlp="select \* from product\_details where product\_details\_id='$product\_details\_id'";

$resultp = mysqli\_query($conn,$sqlp);

$rowp=mysqli\_fetch\_array($resultp,MYSQLI\_ASSOC);

?>

<form action="product\_details\_update.php" method="post" enctype="multipart/form-data" class="form-horizontal" novalidate>

<input type="hidden" placeholder="Product\_details\_id" value="<?php echo $rowp['product\_details\_id']; ?>" name="product\_details\_id" id="product\_details\_id" required="required" />

<div class="form-group item">

<label class="col-md-12">Product\_category\_id</label>

<div class="col-md-12">

<?php

require('db\_connect.php');

$sql="SELECT \* FROM product\_category";

$res=mysqli\_query($conn,$sql);

?>

<select name="product\_category\_id" id="product\_category\_id" class="form-control">

<option value="">--Select Value--</option>

<?php

while ($row = mysqli\_fetch\_array($res,MYSQLI\_ASSOC))

{

?>

<option value="<?php echo $row['product\_category\_id']; ?>"><?php echo $row['product\_category\_name']; ?></option>

<?php

} ?>

</select>

</div>

</div>

<div class="form-group item">

<label class="col-md-12">form\_details\_id</label>

<div class="col-md-12">

<?php

require('db\_connect.php');

$sql1="SELECT \* FROM form\_details";

$res1=mysqli\_query($conn,$sql1);

?>

<select name="form\_details\_id" id="form\_details\_id" class="form-control">

<option value="">--Select Value--</option>

<?php

while ($row1 = mysqli\_fetch\_array($res1,MYSQLI\_ASSOC))

{

?>

<option value="<?php echo $row1['form\_details\_id']; ?>"><?php echo $row1['form\_details\_name']; ?></option>

<?php

}

?>

</select>

</div>

</div>

<div class="form-group item">

<label class="col-md-12">NAME</label>

<div class="col-md-12">

<input type="text" class="form-control" pattern="alpha" placeholder="product\_details\_name" name="product\_details\_name" id="product\_details\_name" required="required" value="<?php echo $rowp['product\_details\_name']; ?>" />

</div>

</div>

<div class="form-group item">

<label class="col-md-12">QUANTITY</label>

<div class="col-md-12">

<input type="text" class="form-control" pattern="numeric" placeholder="product\_details\_quantity" name="product\_details\_quantity" id="product\_details\_quantity" required="required" value="<?php echo $rowp['product\_details\_quantity']; ?>" />

</div>

</div>

<div class="form-group item">

<label class="col-md-12">DESCRIPTION</label><div class="col-md-12">

<input type="text" class="form-control" pattern="alpha" placeholder="product\_details\_description" name="product\_details\_description" id="product\_details\_description" required="required" value="<?php echo $rowp['product\_details\_description']; ?>" />

</div>

</div>

<div class="form-group item">

<label class="col-md-12">PRICE</label>

<div class="col-md-12">

<input type="text" class="form-control" pattern="numeric" placeholder="product\_details\_price" name="product\_details\_price" id="product\_details\_price" required="required" value="<?php echo $rowp['product\_details\_price']; ?>" />

</div>

</div>

<div class="form-group item">

<label class="col-md-12">IMAGE</label>

<div class="col-md-12">

<img src="photos/<?php echo $rowp['product\_details\_image']; ?>" style="width: 100px; height: 100px;">

</div>

</div>

<div class="form-group item">

<label class="col-md-12">NEW IMAGE</label>

<div class="col-md-12">

<input type="hidden" name="oldphoto" id="oldphoto" value="<?php echo $rowp['product\_details\_image']; ?>">

<input type="file" name="product\_details\_image" id="product\_details\_image">

</div>

</div>

<button type="submit" class="btn btn-primary">Save</button>

<input type="button" name="button" class="btn btn-primary" id="button" value="Cancel" onclick="window.location.href='product\_details\_view.php'" />

</form>

</div>

</div>

</div>

</div

<footer class="footer text-center"> 2017 &copy; Elite Admin brought to you by themedesigner.in </footer>

</div>

</div>

<?php include('footer.php');?>

<?php include('valid.php');?>

</body>

</html>

**PRODUCT UPDATE**

<?php

require("db\_connect.php");

$product\_details\_image=$\_FILES["product\_details\_image"]["name"];

if (empty($product\_details\_image))

{

$product\_details\_image=$\_POST['oldphoto'];

}

$product\_details\_id=$\_POST['product\_details\_id'];

$product\_category\_id=$\_POST['product\_category\_id'];

$form\_details\_id=$\_POST['form\_details\_id'];

$product\_details\_name=$\_POST['product\_details\_name'];

$product\_details\_quantity=$\_POST['product\_details\_quantity'];

$product\_details\_description=$\_POST['product\_details\_description'];

$product\_details\_price=$\_POST['product\_details\_price'];

$sql = 'UPDATE `product\_details` SET

`product\_details\_id`='."'$product\_details\_id'".',

`product\_category\_id`='."'$product\_category\_id'".',

`form\_details\_id`='."'$form\_details\_id'".',

`product\_details\_name`='."'$product\_details\_name'".',

`product\_details\_quantity`='."'$product\_details\_quantity'".',

`product\_details\_description`='."'$product\_details\_description'".',

`product\_details\_price`='."'$product\_details\_price'".',

`product\_details\_image`='."'$product\_details\_image'".'

Where product\_details\_id="'.$product\_details\_id.'"';

$result = mysqli\_query($conn,$sql);

if(! $result )

{

echo '<script type="text/javascript">

alert("Record Not Updated");

window.location="product\_details\_view.php";

</script>';

}

else

{

echo '<script type="text/javascript">

alert("Record Updated Successfully");

window.location="product\_details\_view.php";

</script>';

}

?>

**PRODUCT DELETE**

<?php

require("db\_connect.php");

$product\_details\_id=$\_REQUEST['id'];

$sql = 'DELETE FROM `product\_details` Where product\_details\_id="'.$product\_details\_id.'"';

$result = mysqli\_query($conn,$sql);

if(! $result )

{

echo '<script type="text/javascript">

alert("Record Not Deleted");

window.location="product\_details\_view.php";

</script>';

}

else

{

echo '<script type="text/javascript">

alert("Record Deleted Successfully");

window.location="product\_details\_view.php";

</script>';

}

?>

**5. TESTING**

**5.1 Testing Objective:**

Testing is process of executing a program with the intent of finding an error. A good test case is one that high probability of finding an as yet undiscovered error. A successful test is one that uncovers an as yet undiscovered error. The above objectives imply a dramatic change in view point. Testing cannot show the absence of defect, it can only show that software errors are present.

**5.2 Testing Strategies:**

There are two general strategies for testing software. These are as follows

* **Code Testing:**

This examines the logic of the program. To follow this test, cases are developed such that every path of program is tested.

* **Specification Testing:**

Specification Testing examines the specification starting what the program should do and how it should perform under various conditions. Then test cases are developed for each condition and combinations of conditions and to be submitted for processing

**5.3 Testing Method Used:**

Here black box testing and statistical testing are used. In black box testing, all possible types of inputs and seen for corresponding outputs and if not giving, code are corrected. In statistical testing, checking for all variables whether they assigned values before using it, whether array bound correctly defined, whether looping statement terminating without going to infinite loop, whether function parameter are passed in order and about number of parameters etc, are checked successfully and found correct everything working satisfactorily.

**The stages in testing process are**

**Unit Testing:**

Individual components are tested to ensure that they operate correctly. Each component tested independently without other system components.

Ex. Checked for Username and Password with the table, after the next module is loaded session allocation.

**Module Testing:**

Module is collection of dependent components such as an object classes and an abstract data type are some looser collection of procedures and functions. A module encapsulates related components so can be tested without other system modules.

Ex. Valid Blood group with its proper working and with some dummy values is checked. This is checked independent of all other modules.

**Subsystem Testing:**

This phase involves testing collection of modules, which have been integrated into subsystem. Subsystem may be independently designed and implemented. The most common problems, which arise in the large software systems, are subsystems interface mismatches.

Ex. Created separate web pages for user subsystem and these are checked without involvement of other subsystem such as Hostilities, Administrator.

**System Testing:**

The subsystems are integrated to make up the entire system. The testing process is concerned with finding errors, which result from un anticipated interactions between subsystem and system component. It is also concerned with validating that the system is functional and non functional requirements.

Ex. Those all subsystem are integrated and checked for inter-dependency between the subsystems.

**Acceptance Testing**

This is final stage in testing process before the system is tested for operational use. The System is tested with data supplied by system procurer rather than simulated test data. Acceptance testing may also reveal the requirements problems where the system facilitates do not really meet the user’s needs or system performance is unacceptable.

Ex. We tested for all the objectives which were stated in the project statement whether they meet the requirements or not.

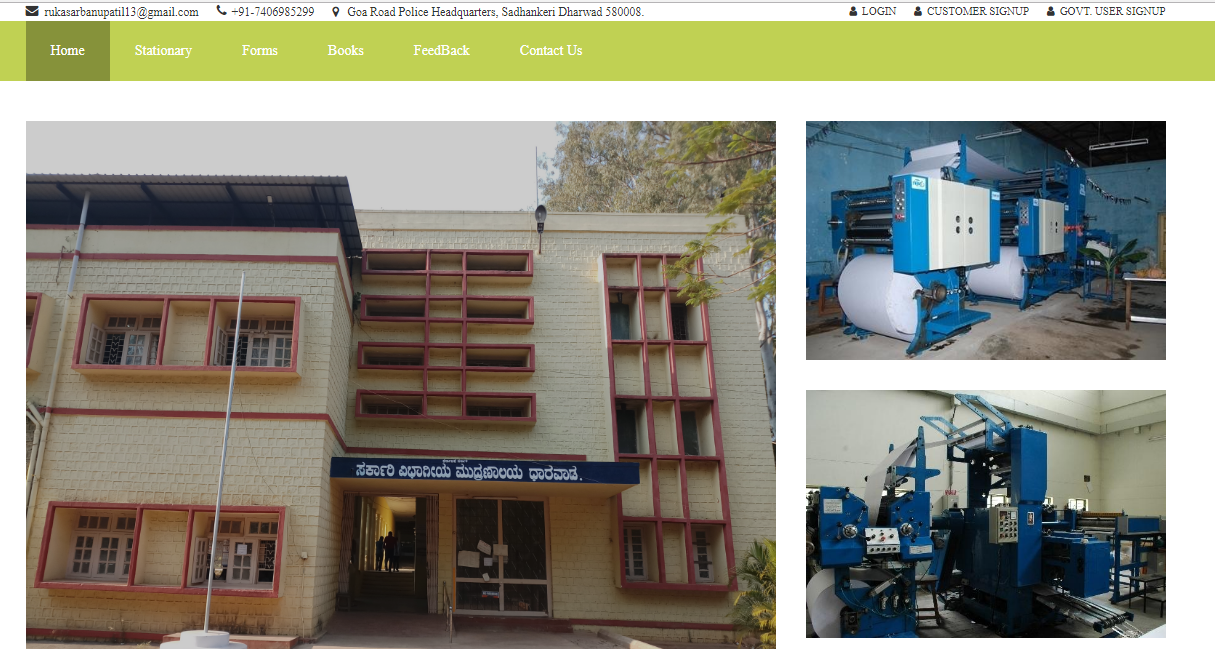
**5.4 Test Cases and Results**

**Test case**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case** | **Input** | **Expected O/P** | **Actual O/P** | **Result** |
| 1 | Valid Username and Password | It should display respective page according to user type. | Respective Home is displayed  . | Passed |
| 2 | Invalid Username and Password | It should give appropriate error message saying “Enter proper User-Id and Password” | Error message Displayed | Passed |
| 3 | Add/Update/Delete Member details | Add/Update/delete action is taken. | Add/Update/delete Member done successfully | Passed |
| 4 | User enters valid Username and password. | Respective Home is displayed | Respective Home is displayed  . | Passed |

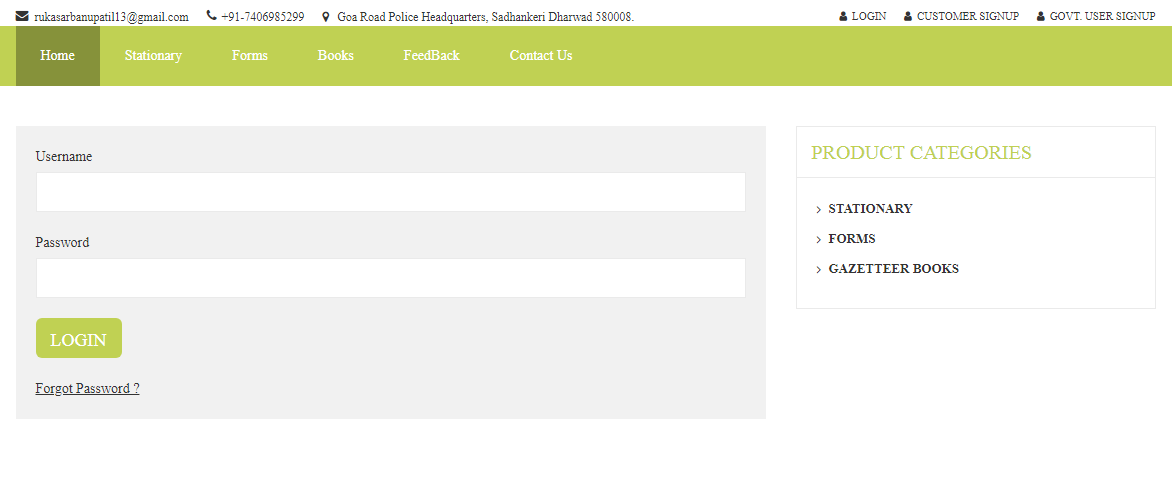
**6. RESULTS AND SCREENSHOTS**

**HOME PAGE**

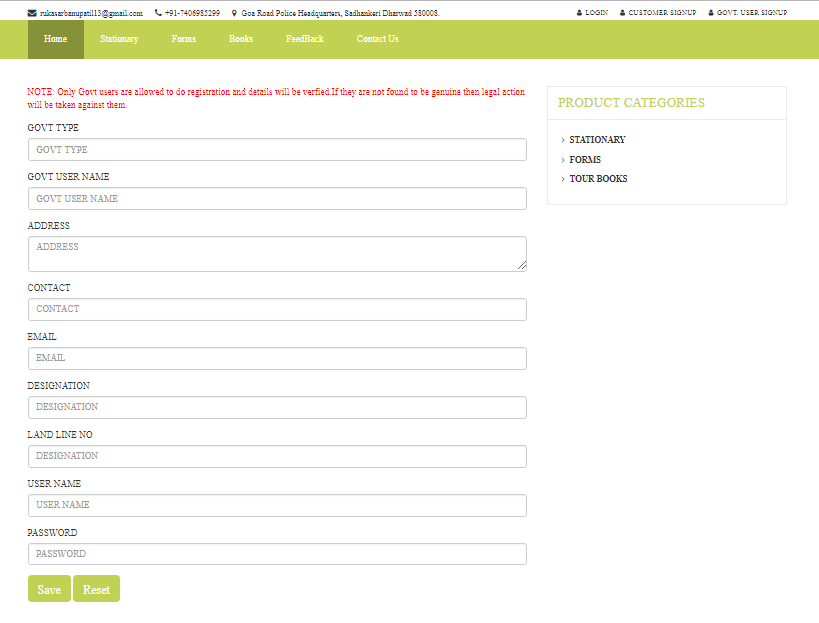
****

**Fig 6.1: Home page of Government Printing Press.**

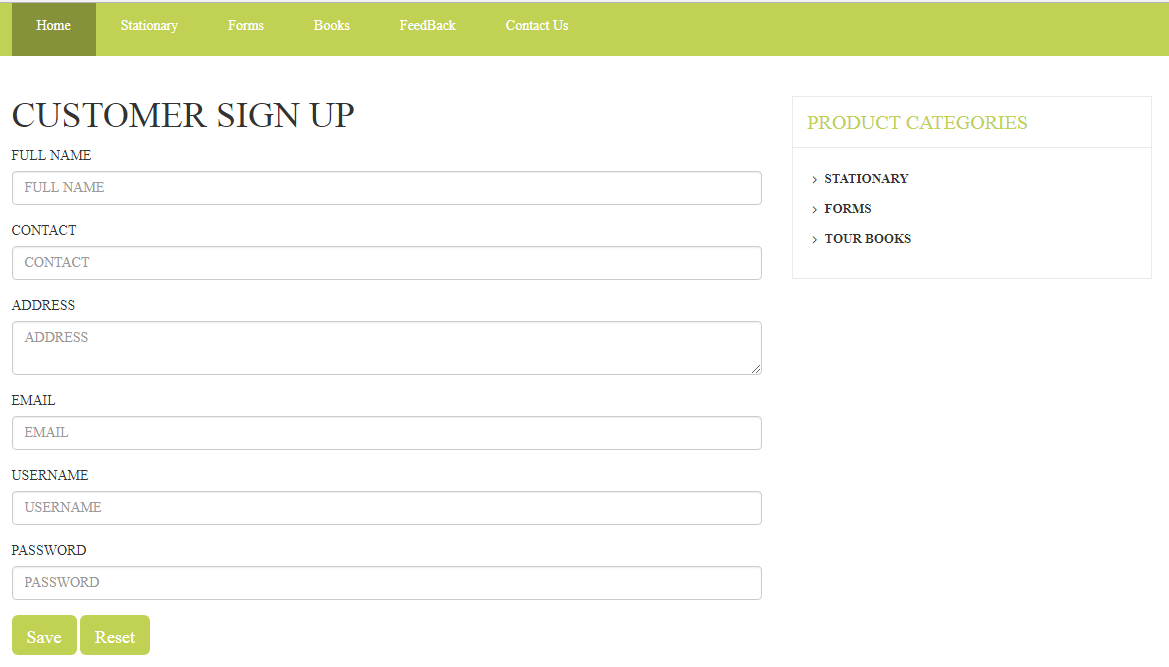
**LOGIN PAGE**

**Fig 6.2: Login page for admin, Govt. user and customer.**

**GOVERNMENT USER REGISTRATION PAGE**

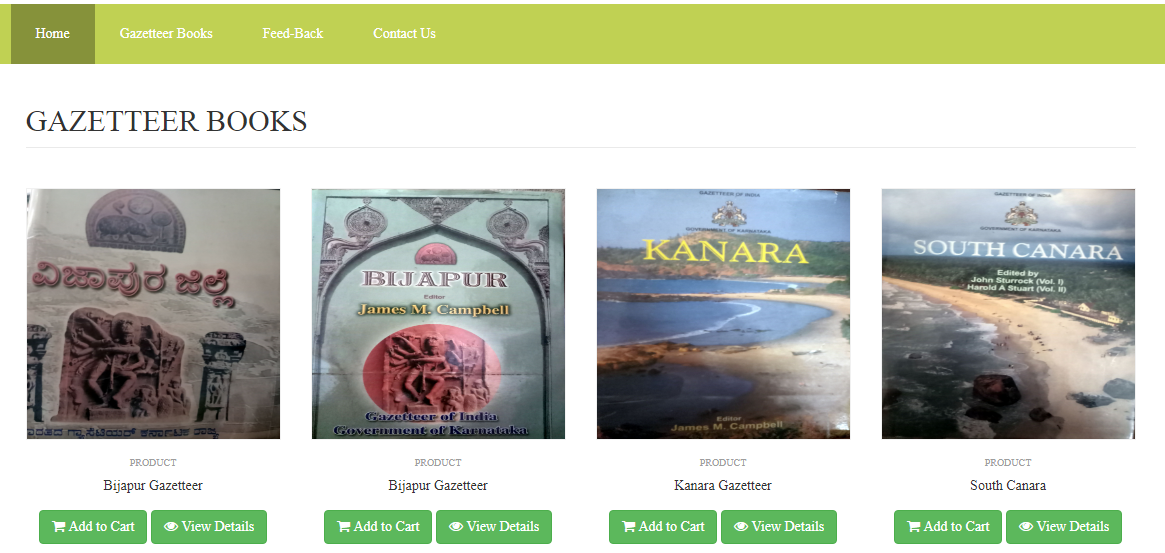
**Fig 6.3: Government user registration form.**

**CUSTOMER REGISTRATION PAGE**

****

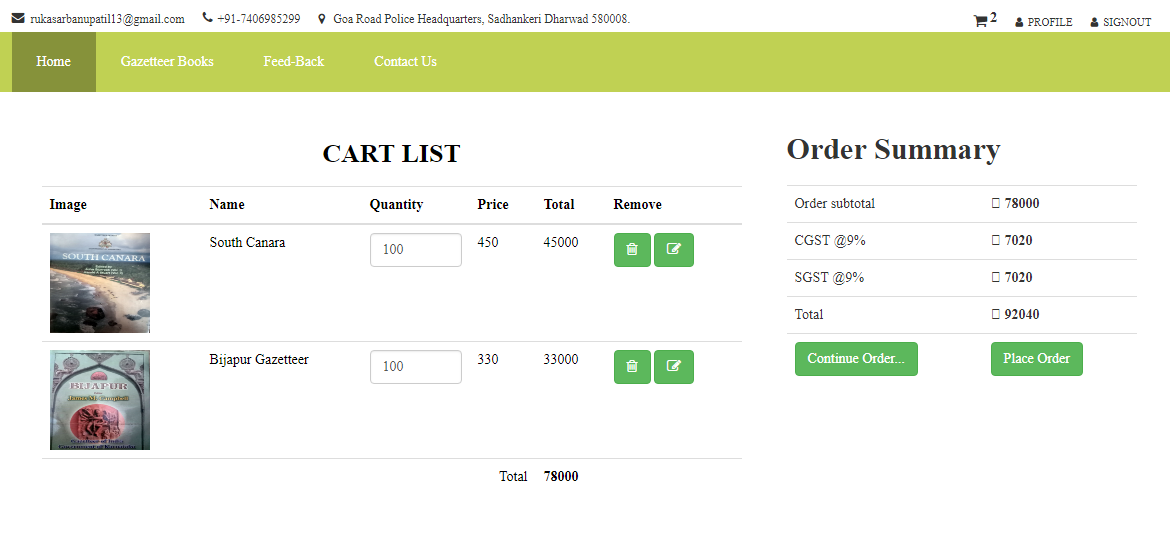
**Fig 6.4: Customer registration form.**

**PRODUCT PAGE**

****

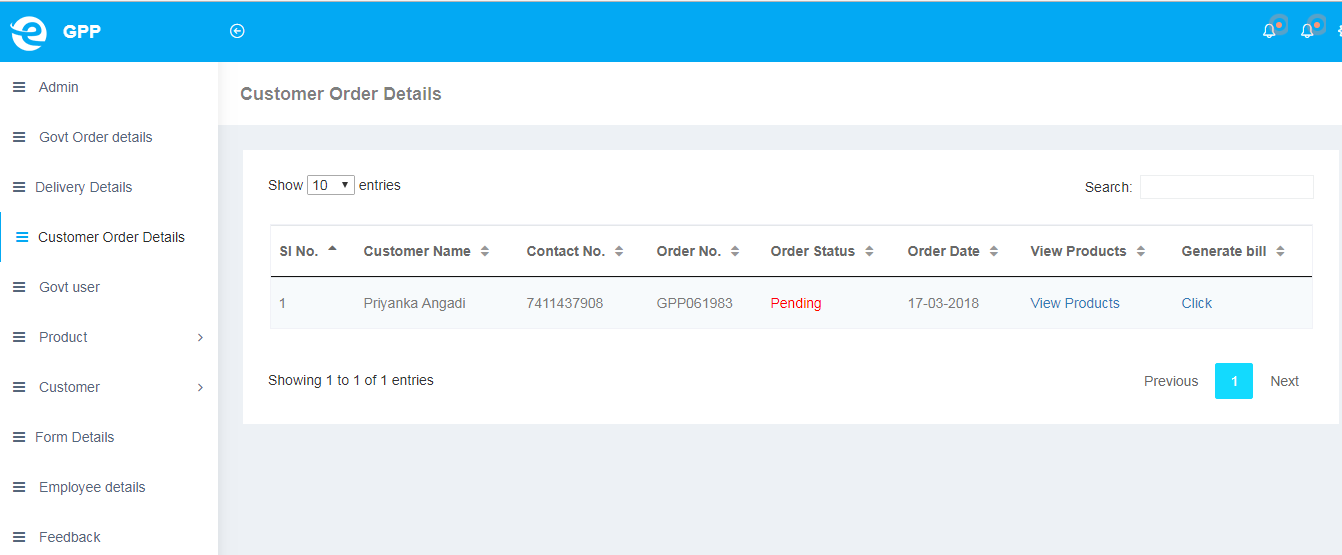
**Fig 6.5: Product page include Gazetteer books, Forms, and Stationary Products.**

**CART PAGE**

****

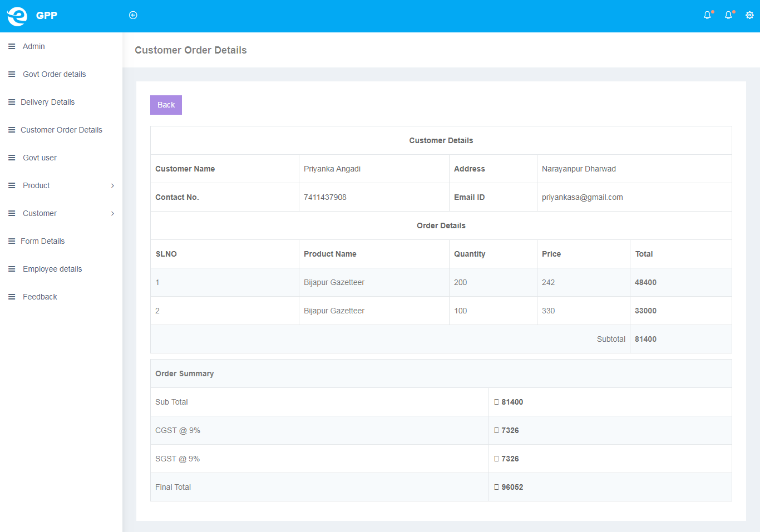
**Fig 6.6: Cart page it contains the products which are ordered by users.**

**CUSTOMER ORDER PAGE**

****

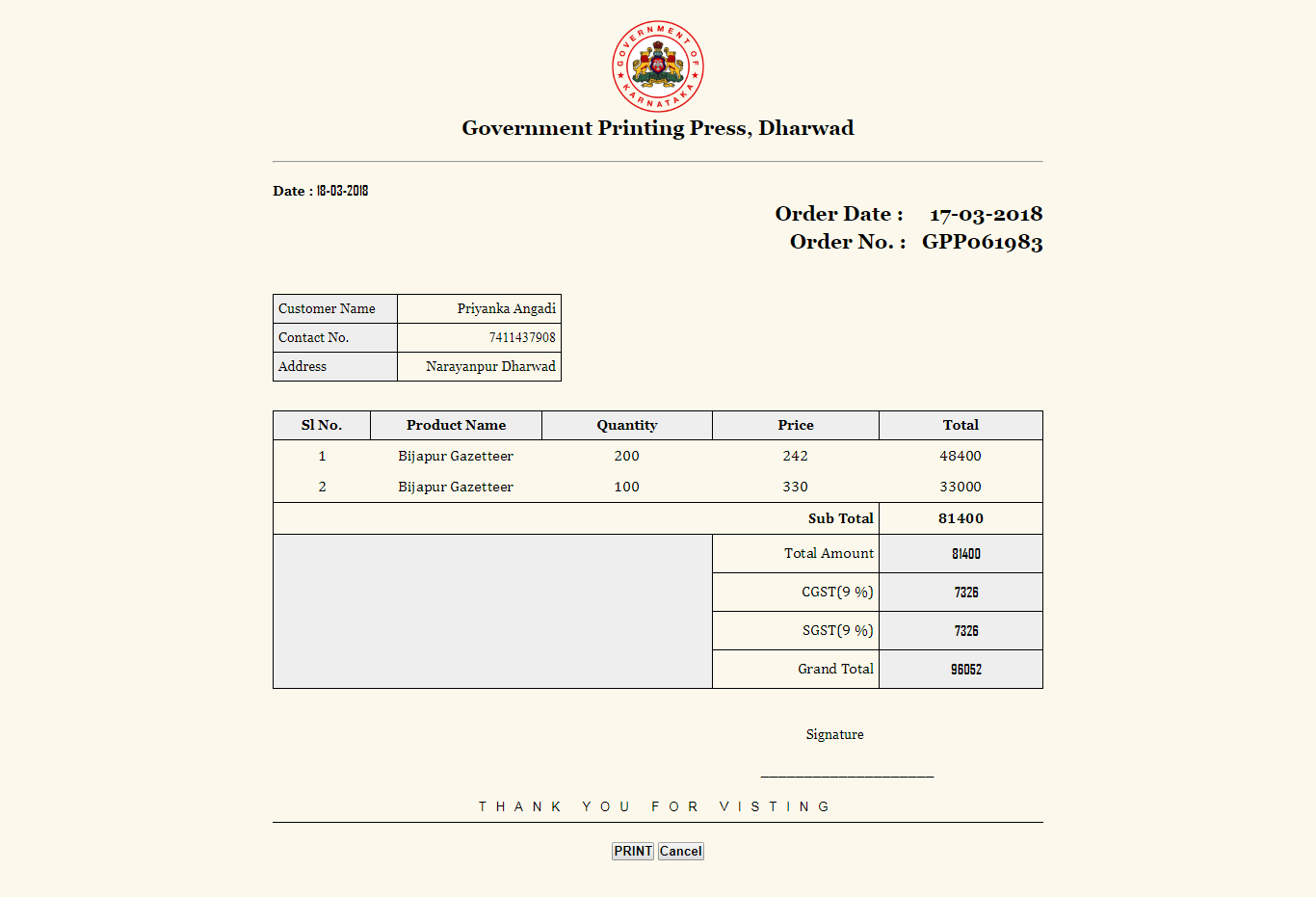
**Fig 6.7: Customer order page, all ordered product are store here.**

**CUSTOMER ORDER DETAILS PAGE**



**Fig 6.8: Customer order details page, all details of ordered product are store here.**

**BILL PAGE**



**Fig 6.9: Bill page here admin can generate bill for customer and Govt. user.**

**7. FUTURE ENHANCEMENT**

* Government office users can make online payment for their work order.
* If they want any other improvement they need in this web application they can improve.
* Customer can make online payment for their order and delivery product to their home itself or their organization.

**CONCLUSION**

* This web application has been developed as per the requirements from ends that is, printing press and government users. Hope this will provide one stop solution for their problems and work efficiently.
* Software is said to have attained its objective only when it meets all the requirements of the end users, the end users can judge the success of the service.
* “GOVERNMENT PRINTING PRESS”service which is web based application which helps to facilitate the office product information between admin, government user and customer.
* The software is efficient, easy to use, has good interface, which reduces the communication gap between system and users.
* The system is feasible enough for amendment and modifications that may arise in future.

**Importance of the system**

* Less manual work.
* Increased efficiency.
* Decreases the rate of errors.
* It reduces the time consumption.
* Quick (instant) result.
* Reduce economy rate.

**8. REFRENCES**

**Websites:**

* <http://php.net/manual/en/language.types.intro.php>
* <https://docs.oracle.com/cd/B19306_01/server.102/b14200.pdf>
* <http://www.w3schools.com>

**Reference Books**

* Software Engineering”, by Ian Somerville, Sixth Edition, Pearson Education Ltd 2007.
* “Web Programming”, by ‘Chris Bates’ Wiley Dreamtech India, 2nd Edition.
* “Data Base Management System” by Steve ‘Shamkant navathe And Ramez elmarsi.