A project report on

BCA-CC-606

“Bookstore Management System”

Submitted to **Smt. K.B. Parekh College of Computer Science-Mahuva**

(Affiliated to Maharaja Krishnakumarsinhji Bhavnagar University)



in partial fulfillment for the award of degree of

BACHELOR

OF

COMPUTER APPLICATIONS

Submitted by

**MAKWANA DHAVAL N. (BCA SEMESTER-6 SEAT NO: 21260256)**

**BARAIYA KUMAR K. (BCA SEMESTER-6 SEAT NO: 21260254)**

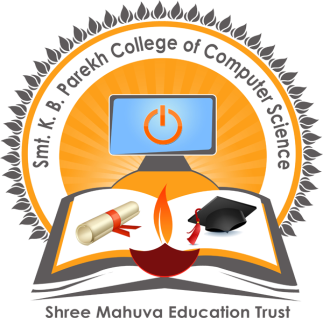
Guided by

**ASHSISH PANDYA**

Assistant Professor

Smt. K.B. Parekh College of Computer Science-Mahuva

**March - 2019**



**Smt.K.B.Parekh College of Computer Science Mahuva-364290**

**(Affiliated to Maharaja Krishnakumarsinhji Bhavnagar University)**

**Date:** 15**/**03/2019

**TO WHOMSOEVER IT MAY CONCERN**

This is to certify that the Student **Makwana Dhaval** and **Baraiya Kumar** of Smt.K.B.Parekh College of Computer Science Mahuva has satisfactorily completed his/her Project **BOOKSTORE MANAGEMENT SYSTEM** during the period December 2019 to March 2019 in the partial fulfillment of BCA-CC-606.

**Name & Signature of Project Guide:**

**Signature and Seal of Principal.**

**Address:** Smt.K.B.Parekh College of Computer Science, Prabhat Nagar Road, Near Parekh College Campus, Cooperative Housing Society, Mahuva, Gujarat 364290 Ph-02844/228332 Email:kbpbcamahuva2000@gmail.com

**ACKNOWLEGEMENT**

We have taken efforts in this project. However, it would not have been possible without the kind support and help of our Faculties. We would like to extend my sincere thanks to all of them.

We are highly indebted to K. B. Parekh College of Computer Science, Mahuva for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would like to express my gratitude towards my parents & member of KBP for their kind co-operation and encouragement which help us in completion of this project. Last but not least, many thanks go to the head of the project, Mr. Ashsish Pandya whose have invested his full effort in guiding the team in achieving the goal. We have to appreciate the guidance given by other supervisor as well as the panels especially in our project presentation that has improved our presentation skills thanks to their comment and advices.

We would like to express our special gratitude and thanks to all above mentioned people for giving us such attention and time. Our thanks and appreciations also go to our colleague in developing the project and people who have willingly helped us out with their abilities.

ABSTRACT

Bookstore Management System is basically used to build an web application program which help people to find and buy latest design of Books with different categories like Biography, Programming, Management, etc. It is useful in the way that it makes an easier way to buy Personal Bookstore.

Today most of the book shop is useful for shopping site. The admin have lots of paper work and they are using desktop, spread sheet like MS Excel application to manage data in soft copy about user record. In this proposed Bookstore System it will run in server and user can handle whole the registration activities.

This application maintains the centralized database so that any changes done at a location reflects immediately. This is an online tool so more than one user can login into system and use the tool simultaneously.

The aim of this application is to reduce the manual effort needed to manage transactions and historical data used in various gods owns. Also this application provides an interface to users to view the details And Design about Bookstore.

# Table of Contents

[INTRODUCTION 8](#_Toc3818959)

[1.1 Project Background 9](#_Toc3818960)

[1.2 Objectives of project 9](#_Toc3818961)

[1.3 Purpose of Project 9](#_Toc3818962)

[1.4 Scope of Project 10](#_Toc3818963)

[1.5 Applicability of Project: 10](#_Toc3818964)

[REQUIREMENT AND ANALYSIS 11](#_Toc3818965)

[2.1 Problem Statement 12](#_Toc3818966)

[2.2 Requirement specifications](#_Toc3818967) 12

[2.3 Hardware requirement 13](#_Toc3818968)

[2.4 Software Requirement 13](#_Toc3818969)

[2.5 Planning and scheduling 14](#_Toc3818970)

[SYSTEM DESIGN 23](#_Toc3818971)

[3.1 Over All System Design Using Designing Tools 24](#_Toc3818972)

[3.2 Data dictionary 37](#_Toc3818973)

[3.3 Input/Output design 41](#_Toc3818974)

[TESTING AND IMPLEMENTATION 56](#_Toc3818975)

[4.1 Testing Approach Used 57](#_Toc3818976)

[4.2 Test Cases 59](#_Toc3818977)

[4.3 Implementation approaches 63](#_Toc3818978)

[CONCLUSION 64](#_Toc3818979)

[5.1 Limitation of system 65](#_Toc3818980)

[5.2 Future Scope of System 66](#_Toc3818981)

[5.3 Bibliography 66](#_Toc3818982)

# Contents of figures

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Figure Name** | **Page No** |
| 1 | Planning and Scheduling | 14 |
| 2 | Spiral Model | 20 |
| 3 | Data Flow Diagram Symbols | 25 |
| 4 | 0 Level Data Flow Diagram | 25 |
| 5 | 1st Level Data Flow Diagram | 26 |
| 6 | BMS Flowchart Diagram | 26 |
| 7 | User Flow Diagram | 27 |
| 8 | Use Case Diagram Symbols | 28 |
| 9 | User Use Case Diagram | 29 |
| 10 | BMS Use Case Diagram | 30 |
| 11 | Activity Diagram Symbols | 32 |
| 12 | User Activity Diagram | 33 |
| 13 | Login System Activity Diagram | 34 |
| 14 | E-R Diagram Symbols | 35 |
| 15 | E-R Diagram for Bookstore Management System | 36 |
| 16 | BMS Home Page | 41 |
| 17 | BMS Selected Category | 42 |
| 18 | BMS Book Details | 43 |
| 19 | BMS Login Page | 44 |
| 20 | BMS Register Page | 45 |
| 21 | BMS Contact Us Page | 46 |
| 22 | BMS Cart Page Viewers | 46 |
| 23 | BMS Order Page | 47 |
| 24 | Logged in Home Page | 48 |
| 25 | BMS Users Book Details | 49 |
| 26 | BMS Users Cart Page | 50 |
| 27 | BMS Search Books | 51 |
| 28 | BMS Admin Login Page | 51 |
| 29 | BMS Admin Home Page | 52 |
| 30 | BMS Add New Category | 52 |
| 31 | BMS View Category | 53 |
| 32 | BMS Add New Books | 53 |
| 33 | BMS View Books | 54 |
| 34 | BMS Contacted List Books | 54 |
| 35 | BMS Users List | 55 |
| 36 | BMS Forget Password Page | 55 |
| 37 | Black Box Testing | 57 |
| 38 | Gray Box Testing | 59 |
| 39 | Test Cases 1 | 60 |
| 40 | Test Cases 2 | 61 |
| 41 | Test Cases 3 | 61 |
| 42 | Test Cases 4 | 62 |

**Chapter 1**

**Introduction**

# Project Background

* This Software allows the Admin to store the book details and the customer details.
* Easier access to information like customer information and availability.
* Provide facility of storing data to reduce the paper work.
* In Bookstore Management System Users can by a book and Admin shows their name and other background of the user.
* A new idea about Project how Bookstore Management System works.

For make a system computerized.

# Objectives of Project

* To reduce the paper work.To make computerized system.
* Increase operational speed.Faster searching as well as accuracy.
* Large storage of data using database.
* Manual process of vehicle purchase and sales on finance and cash and generate reports of model wise, weekly, monthly ,annual progress is so difficult so this project make is easier.
* Speed and faster information retrieval.

# 

# 1.3 Purpose of Project

* The main purpose of Book-store Management System is to focused on the solution of all the problems related to the paper work from the different reasons.
* It provides a facility to handle all the activities at one place. With the help of this application, admin can perform different kind of operations at the same time and place.
* Bookstore management System has an ability to keep the records safe related to Books.
* We provide the best service in our website or focuses on user choice. We will improve new feat user can easily understand and trust our system.

# 1.4 Scope of Project

* The intentions of the system are to reduce over-time pay and increase the number of records that can be treated accurately; Requirements statements in this document are both functional and non-functional.
* Correct and Accurate Searching that provides the result by applying search operation.
* Customers can book a book with just few clicks.
* Give flexibility admin to use database effectively and utilize the word, not pad and calculator Unambiguous and understandable by all level facilities effectively.
* Unambiguous and understandable by all level.

# 1.5 Applicability of Project:

* For customers who want to buy books at anywhere or anytime.
* Admin applicable for insert books, list of books.
* Database is used for store and fetch data from or to the database so both users and admin can fetch or read data.

# Chapter 2

# Requirement And Analysis

# 2.1 Problem Statement

* So much Paper work
* Process is much time consuming
* Extra expense in paper work
* Large storage of data
* Manual process of Vehicle purchase and sales on finance and cash and generate reports of model wise, weekly, monthly, annual progress is so difficulties.
* Speed and faster information retrieval.
* Accuracy and consistency in manual system is less.
* Personal delay.
* In manual system it is tedious task to search a particular record later after.
* Increase the staff in test taken place and wasting of their precious time.

# 2.2 Requirement Specification

As per the System Requirements it contains two (2) Modules:

1) Admin

2) Client

### Functionalities of Admin:

* This Module includes the mainly following tasks:
  + Entry of Category.
  + Category List.
  + Add a New Book.
  + View Book.
  + View Message which Send by Client.

### Functionalities of Client:

* This Module includes the mainly following tasks:
  + View Books.
  + Add books to Cart.
  + Search Books.
  + View or Add items in Cart.

# Hardware requirement

* System type 32 bit Operating System.
* Windows 7/8/8.1/10
* Linux Ubuntu / Light ubuntu
* Mac OS
* 350MB RAM

# Software Requirement

* Wamp Server
* MySQL
* Browser
* PHPMyAdmin

#### Client Side Tools

* **Processor :** PC with a Dual core processor or above isϖ Recommended: 2.20 GHz processor.
* **RAM :** 512 MB or onwards Recommended.
* **Hard Disk** **:** 45 MB of available space required on system drive of available or more.
* **Operating System :** Windows or open source 32/64 bit operating system, or later versions. Browser Mozilla Firefox 2.0 /Internet Explorer 8.0 Onwards / Googleϖ Chrome.

# Planning and scheduling

Different amount of time may be required for each stage in the project cycle, depending on the particulars of the key aspect of the project cycle seem to recur during development process. The information obtained during the requirement gathering of pre-development phase provides the impetus for the requirement analysis and the information is further used in the design phase.

**Planning and Scheduling**:**-**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Task Name** | **Start/Finish** | **Duration** |
| 1 | Analysis | 25/12/2018 to 01/01/2019 | 8 Days |
| 2 | Design | 01/01/2019 to 09/01/2019 | 9 Days |
| 3 | Coding | 10/01/2019 to 08/02/2019 | 4 Weeks |
| 4 | Implementation | 08/02/2019 to 12/02/2019 | 5 Days |
| 5 | Testing | 12/02/2019 to 17/02/2019 | 6 Days |
| 6 | Documentation | 18/02/2019 to 10/3/2019 | 3Weeks |

### (Figure 1 : Planning and Sheduling)

The above schedule specifies the estimated time that will be required in various software development phases, considering all situational factors. Team members are technically ready accepting few days training on to get the Technology Awareness. Thus, according to calculation, it is feasible to build such solution in time. “**The schedule will be revised at the end of each phase and updated as required”.**

**Brief overview of the technology**

**Front End - HTML, CSS , BOOTSTRAP**

1. **HTML**

HTML stands for HYPER TEXT MARKUP LANGUAGE, which is most¬ widely used language on web to develop web pages. HTML refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a web page is called Hypertext.

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.

As its name suggests, HTML is a Mark-up Language which means you use HTML to simply “mark-up” a text document with tags that tells a web browser how to structure it to display.

Originally, HTML was develop with the intent of defining the structure of documents like heading, paragraph, lists, and so forth to facilitate the sharing of scientific information between researchers. Now, HTML is being widely used to format web pages with the help of different tags available in HTML.

1. **CSS**

Cascading Style Sheet is a style sheet language used for describing the presentation of a document written in a markup language Although most often used to set the visual style of web page and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents

CSS has a simple syntax and uses a number of English keywords to specify the names of various style properties. A style sheet consists of a list of rules. Each rule or rule-set consists of one or more selectors, and a declaration block.

1. **BOOTSTRAP**

Bootstrap is a free and open-source, front-end web frame work for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only.

Bootstrap is modular and consists of a series of less stylesheets that implement the various components of the toolkit. These stylesheets are generally compiled into a bundle and included in web pages, but individual components can be included or removed. Bootstrap provides a number of configuration variables that control things such as color and padding of various components.

Since Bootstrap 2, the Bootstrap documentation has included a customization wizard which generates a customized version of Bootstrap based on the requested components and various settings.

As of Bootstrap 4, is used instead of less for the stylesheets. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code.

**Back End - PHP, MySQL**

1. **PHP**

The PHP Hypertext Pre-processor (PHP) is a programming language that allows web developers to create dynamic content that interacts with databases. PHP is basically used for developing web based software applications. This tutorial helps you to build your base with PHP. PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.

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

1. MySQL

MySQL is a database, widely used for accessing querying, updating, and managing data in databases.

MySQL is an open source RDBMS that relies on SQL for processing the data in database. MySQL provides APIs for the languages like C, C++, Eiffel, JAVA, Perl, PHP and Python. MySQL is most commonly used for web applications and for embedded applications and has become a popular alternative to proprietary database system because of its speed and reliability. MySQL can run on UNIX, Windows and Mac OS.

**Project Analysis and Planning**

The BMS is critical to set-up online order, customers to browse through book categories**.** This is a small scale project for BMS. The basic idea is that the customers can buy a book from anywhere during any time by the cash through.

**User**

* User can Register, Login, Logout the system.
* View different categories and by books.
* Contact with Admin
* Add Books to Cart
* Order Books

**Functionality**

* One or more user visit web page at a time.

**Usability**

* In any browser run this webpage.

**Performance**

* It performs the webpage as per User’s operating system.

**Admin**

* Admin can manage system.
* Provide books.

**Functionality**

* Admin can insert a book or manage the records.

**Spiral Model**

The spiral model combines the idea of iterative development with the systematic, controlled aspects of the waterfall model. This Spiral model is a combination of iterative development process model and sequential linear development model i.e. the waterfall model with a very high emphasis on risk analysis. It allows incremental releases of the product or incremental refinement through each iteration around the spiral.

**Spiral Model - Design**

The spiral model has four phases. A software project repeatedly passes through these phases in iterations called Spirals.

**Identification**

This phase starts with gathering the business requirements in the baseline spiral. In the subsequent spirals as the product matures, identification of system requirements, subsystem requirements and unit requirements are all done in this phase.

This phase also includes understanding the system requirements by continuous communication between the customer and the system analyst. At the end of the spiral, the product is deployed in the identified market.

**Design**

The Design phase starts with the conceptual design in the baseline spiral and involves architectural design, logical design of modules, physical product design and the final design in the subsequent spirals.

**Construct or Build**

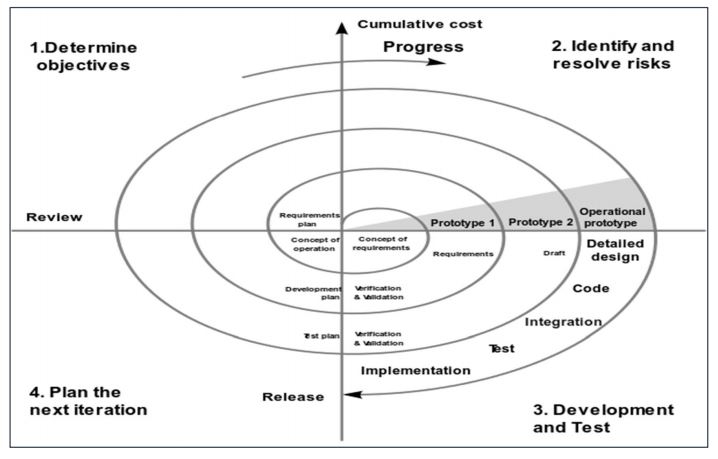
The Construct phase refers to production of the actual software product at every spiral. In the baseline spiral, when the product is just thought of and the design is being developed a POC (Proof of Concept) is developed in this phase to get customer feedback.

Then in the subsequent spirals with higher clarity on requirements and design details a working model of the software called build is produced with a version number. These builds are sent to the customer for feedback.

**Evaluation and Risk Analysis**

Risk Analysis includes identifying, estimating and monitoring the technical feasibility and management risks, such as schedule slippage and cost overrun. After testing the build, at the end of first iteration, the customer evaluates the software and provides feedback.

The following illustration is a representation of the Spiral Model, listing the activities in each phase.



**(Figure 2 : Spiral Model)**

Based on the customer evaluation, the software development process enters the next iteration and subsequently follows the linear approach to implement the feedback suggested by the customer. The process of iterations along the spiral continues throughout the life of the software.

**Spiral Model Application**

The Spiral Model is widely used in the software industry as it is in sync with the natural development process of any product, i.e. learning with maturity which involves minimum risk for the customer as well as the development firms.

The following pointers explain the typical uses of a Spiral Model −

* When there is a budget constraint and risk evaluation is important.
* For medium to high-risk projects.
* Long-term project commitment because of potential changes to economic priorities as the requirements change with time.
* Customer is not sure of their requirements which are usually the case.
* Requirements are complex and need evaluation to get clarity.
* New product line which should be released in phases to get enough customer feedback.
* Significant changes are expected in the product during the development cycle.

**Spiral Model - Pros and Cons**

The advantage of spiral lifecycle model is that it allows elements of the product to be added in, when they become available or known. This assures that there is no conflict with previous requirements and design.

This method is consistent with approaches that have multiple software builds and releases which allows making an orderly transition to a maintenance activity. Another positive aspect of this method is that the spiral model forces an early user involvement in the system development effort.

On the other side, it takes a very strict management to complete such products and there is a risk of running the spiral in an indefinite loop. So, the discipline of change and the extent of taking change requests are very important to develop and deploy the product successfully.

The advantages of the Spiral SDLC Model are as follows −

1. Changing requirements can be accommodated.
2. Allows extensive use of prototypes.
3. Requirements can be captured more accurately.
4. Users see the system early.
5. Development can be divided into smaller parts and the risky parts can be developed earlier which helps in better risk management.

# Chapter 3

# System Design

# 3.1 Over All System Design Using Designing Tools

The Purpose of Design Phase is to plan a solution for problem specified by the requirements. System Design aims to identify the modules that should be in the system, the specification of those modules and how they interact with other to produce the results. The goal of the design process is to produce a model that can be used later to build that system. The produced model is called design of the system.

System Design is the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements.

Normally, the design proceeds in 2 stages:

* Physical Design
* Database Design

### Physical Design

The Physical Design is a graphical representation of a system showing the system’s internal and external entities and the flow of data into and out of these entities. An internal entity is an entity within the system that transforms data.

To represent the Physical Design of the system, we use diagrams like Data Flow Diagrams, E-R Diagrams, Use Case Diagrams, etc…

### Data Flow Diagram

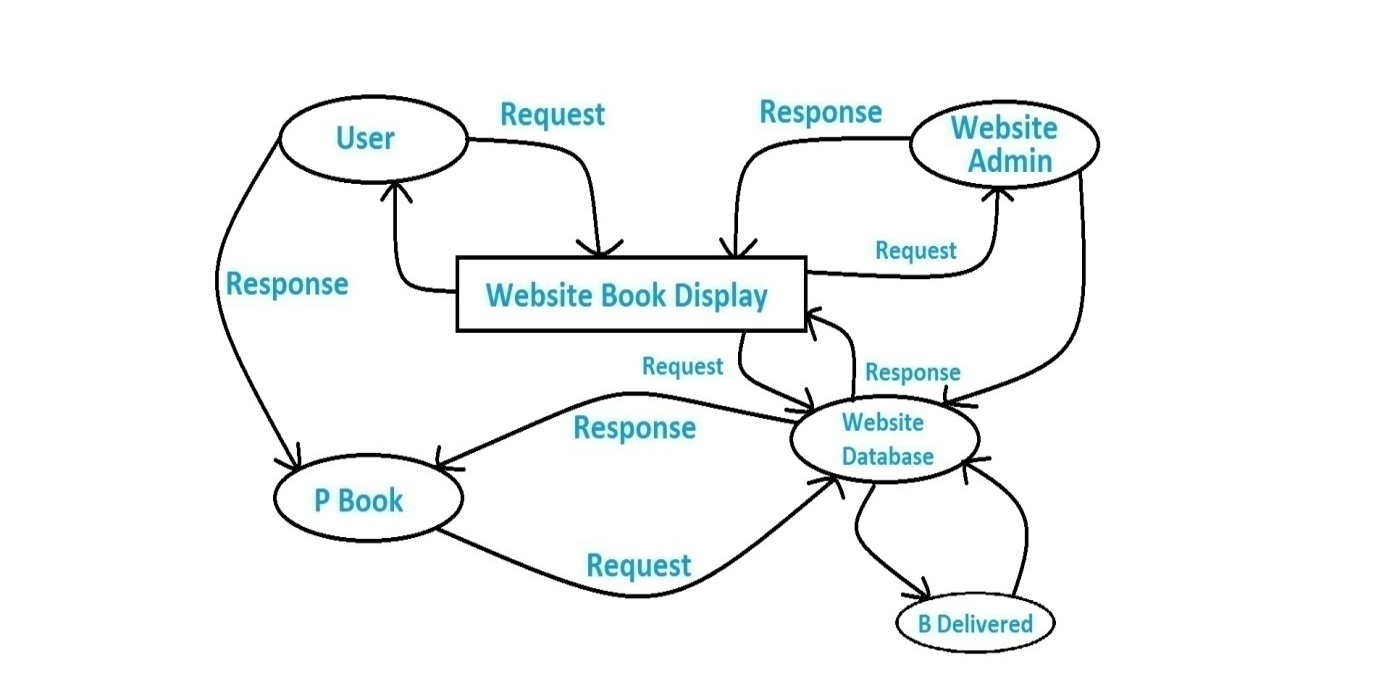
The Data Flow Diagrams (DFD) is a graphical representation of the flow of data through an information system. Data Flow Diagrams are used by systems analysis to design information processing systems but also a way to model whole organization. The main merit of DFD is that it can provide an overview of what data a system would processes. What transformations of data are done, what data are stored and which stored data is used, and where the result is flow.

### Standard Symbols used in DFD:

|  |  |  |
| --- | --- | --- |
| **Symbol** | **Name** | **Function** |
|  | **Data Flow** | Used to connect processes to each other. The Arrowhead indicates the direction of Data Flow. |
|  | **Process** | Performs some transformation to input data to output data. |
|  | **Input / Output** | It is used to Input or Output data. |

### (Figure 3 : Data Flow Diagram Symbols)

### 0 Level DFD (Website Flow Diagram)



### (Figure 4 : 0 Level Data Flow Diagram)

### 1st Level DFD (Website Flow Diagram)

USER

Selection of

Books

Required

information

Books

Select from

books

Add to

cart

Processing

Data

-

Order Successfullylow Diagram

Order Details

List of

books

Select from

books

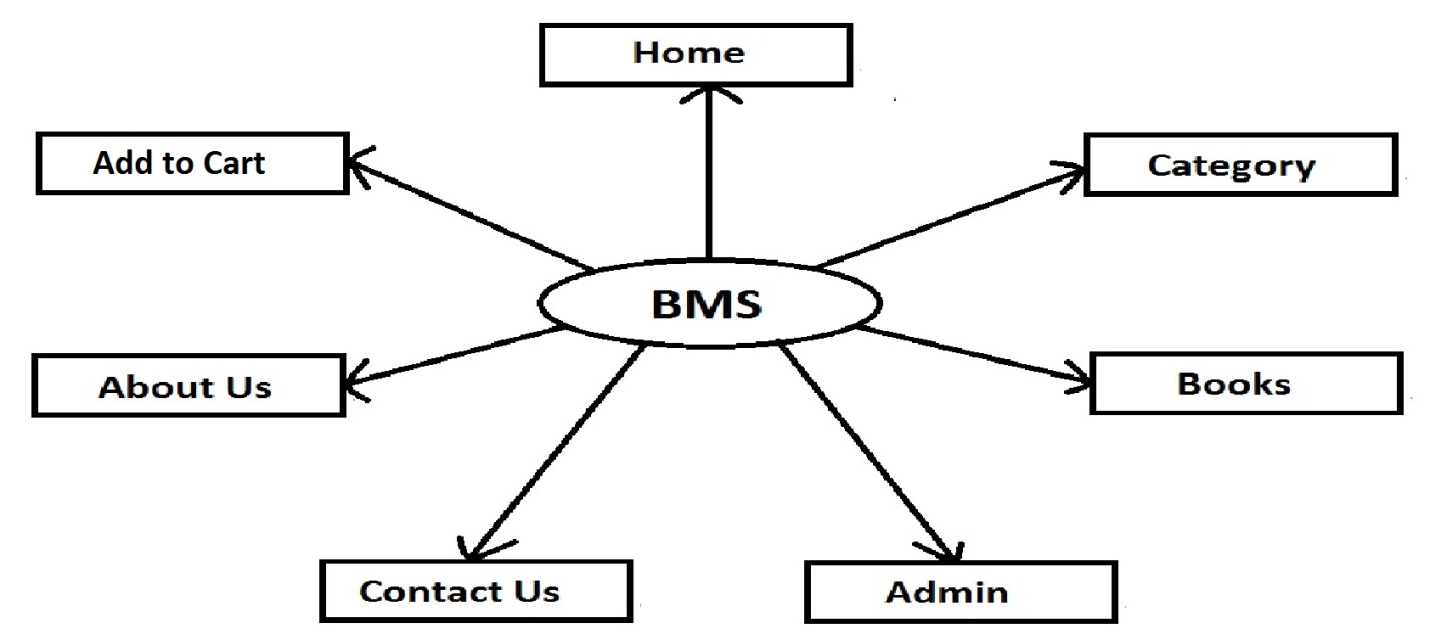
User Details

Order Details

Category

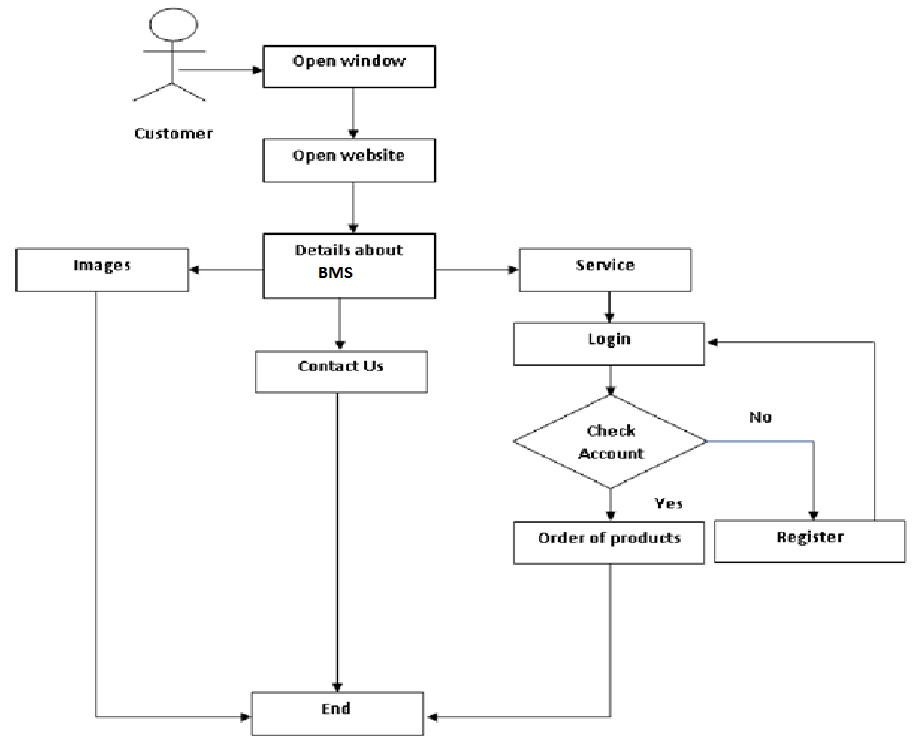
(Figure 5 : 1st Level Data Flow Diagram)

### Flowchart Diagram



(Figure 6 : BMS Flowchart Diagram)

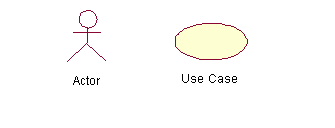
### User Flow Diagram



### (Figure 7 : User Flow Diagram)

# Use Case Diagram

A use case is a set of scenarios that describing an interaction between a user and a system.  A use case diagram displays the relationship among actors and use cases.  The two main components of a use case diagram are use cases and actors.

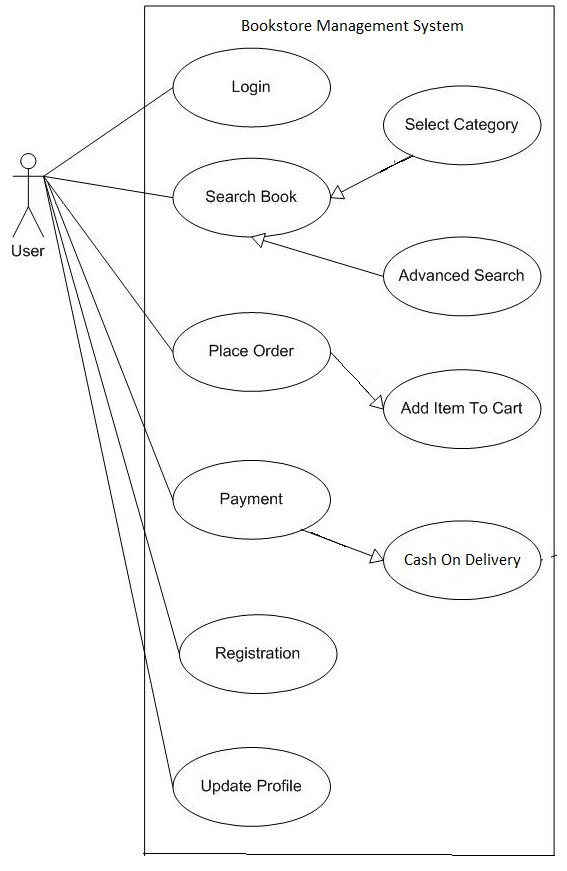


### (Figure 8 : Use Case Diagram Symbols)

An actor is represents a user or another system that will interact with the sys

Item you are modeling.  A use case is an external view of the system that represents some action the user might perform in order to complete a task.

### User Use Case Diagram



### (Figure 9 : User Use Case Diagram)

### BMS Use Case Diagram

BOOKSTORE MANAGEMENT SYSTEM

User

Visitor

Admin

### (Figure 10 : BMS Use Case Diagram)

### Activity Diagram

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

|  |  |  |
| --- | --- | --- |
| **Name of symbols** | **Symbols** | **Description** |
| **Start symbol** | circle.jpg | Represents the beginning of a process or workflow in an activity diagram. It can be used by itself or with a note symbol that explains the starting point. |
| **Activity symbol** | **circle.jpg** | Indicates the activities that make up a modeled process. These symbols, which include short descriptions within the shape, are the main building blocks of an activity diagram. |
| **Connector symbol** |  | Shows the directional flow, or control flow, of the activity. An incoming arrow starts a step of an activity; once the step is completed, the flow continues with the outgoing arrow. |
| **Decision symbol** | **circle.jpg** | Represents a decision and always has at least two paths branching out with condition text to allow users to view options. This symbol represents the branching or merging of various flows with the symbol acting as a frame or container. |
| **End symbol** | **circle.jpg** | Marks the end state of an activity and represents the completion of all flows of a process. |
| **Joint symbol/ Synchronization bar** | circle.jpg | Combines two concurrent activities and re-introduces them to a flow where only one activity occurs at a time. Represented with a thick vertical or horizontal line. |
| **Fork symbol** | **circle.jpg** | Splits a single activity flow into two concurrent activities. Symbolized with multiple arrowed lines from a join. |

### (Figure 11 : Activity Diagram Symbols)

### User Activity Diagram

### (Figure 12 : User Activity Diagram)

### Login System Activity Diagram



### (Figure 13 : Login System Activity Diagram)

### E-R Diagram

Entity-Relationship Diagram is a graphical representation of entities and their relationship to each other. It describes how data is related to each other. An entity is a piece of data- an object or a concept about which data is stored. A relationship is how the data is shared between entities.

In E-R Diagram, there are 3 main components:

|  |  |  |
| --- | --- | --- |
| **Symbol** | **Name** | **Description** |
|  | **Entity** | An Entity can be any object, place, person or anything. |
|  | **Attribute** | An Attribute describes a property or characteristics of an entity |
|  | **Relationship** | A Relationship describes relation between entities. |

### (Figure 14 : E-R Diagram Symbols)

# E-R Diagram for Bookstore Management System

# (Figure 15 : E-R Diagram for Bookstore Management System)

BMS Contains

Cart

Books

Category

Has Many

Customers

Buys

# 3.2 Data Dictionary

### Database Design & Structure Design

Various tables used in the System are as follows:

1. Admin
2. Book
3. Category
4. Contact
5. Register
6. Order

Detail of all the tables with its all the fields are as below:

#### Table Name : Admin

**Primary Key :** a\_id

**Description :** For store Admin login Detail.

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| A\_id | int(4) | Used to Store Admin ID |
| A\_unm | Varchar(3) | Used to Store Username of Admin |
| A\_pwd | Varchar(30) | Used to Store Password of Admin |

#### Table Name : Book

**Primary Key :** b\_id

**Description :** Store Book Details.

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| B\_id | Int(10) | Used to Store Book ID |
| B\_nm | Varchar(50) | Used to Store Book Name |
| B\_cat | Int(6) | Used to Select or Store the Book ID of Different Categories |
| B\_desc | Longtext | Used to Store The Description of The Book in Large Amount Data |
| B\_price | Int(4) | Used to Store the Price of Book |
| B\_img | Varchar(50) | Used to Store the Image Name of Book |
| B\_time | Int(20) | Used to store the Time of Inserted book |

#### Table Name : Category

**Primary Key :** cat\_id

**Description :** Store Category Names.

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| Cat\_id | Int(10) | Used to Store the Category ID |
| Cat\_nm | Varchar(50) | Used to Store the Category Name |

#### Table Name : Contact

**Primary Key :** c\_id

**Description :** Store details for Contact Us.

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| C\_id | Int(4) | Store Contact ID of Client/User |
| C\_fnm | Varchar(100) | Store Full Name of User |
| C\_mno | Int(10) | Store Mobile Number of Client/User |
| C\_email | Varchar(60) | Store the E-Mail Address of Client/User |
| C\_msg | Longtext | Store The Message or Query of The Client/User |
| C\_time | Varchar(20) | Store The Time of Contact Page inserted The Data |

#### Table Name : Register

**Primary Key :** r\_id

**Description :** Details for Registration Visitors or Users.

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| R\_id | Int(8) | Store User Registration ID |
| R\_fnm | Varchar(100) | Store Full Name of User |
| R\_unm | Varchar(50) | Store Username of User |
| R\_pwd | Varchar(30) | Store the Password of User |
| R\_cno | Varchar(10) | Store the Contact Number of User |
| R\_email | Varchar(60) | store E-Mail Address of User |
| R\_time | Varchar(20) | Store Time of Registration of User |

#### Table Name : Order

#### Primary Key : o\_id

#### Description : Details for Order

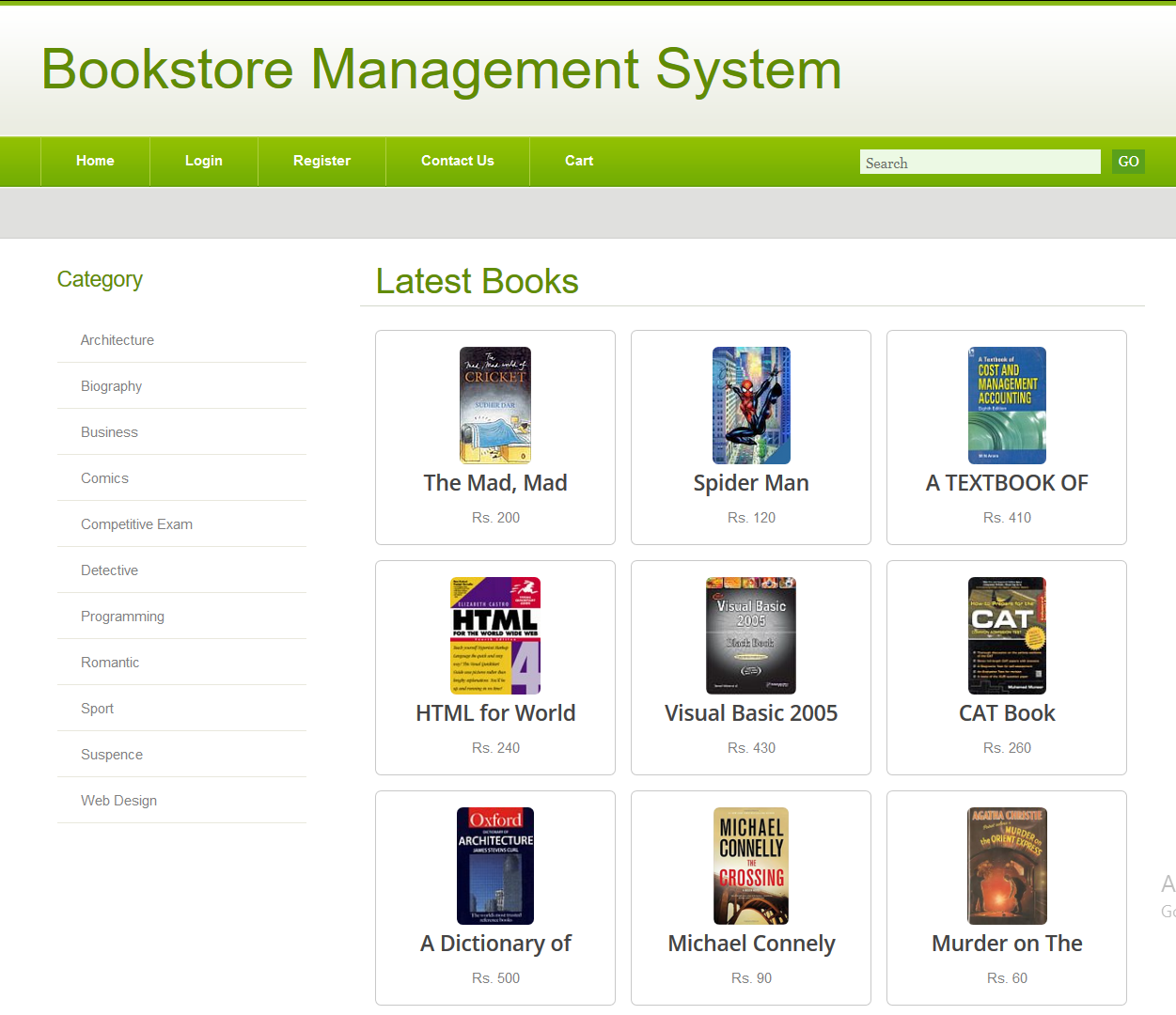
#### .

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| O\_id | Int(11) | Store order ID |
| O\_name | Varchar(30) | Store Full Name of User |
| O\_address | Varchar(200) | Store address of User |
| O\_[pincode | Int(20) | Store city Pincode |
| O\_city | Varchar(30) | Store the City Name |
| O\_state | Varchar(30) | store State |
| O\_mobile | Bigint(20) | Store Mobile Number |
| O\_rid | Int(8) | Store Register ID |

# 3.3 Input/Output Design

1. **Home Page**

Home Page of BMS without logged in User.



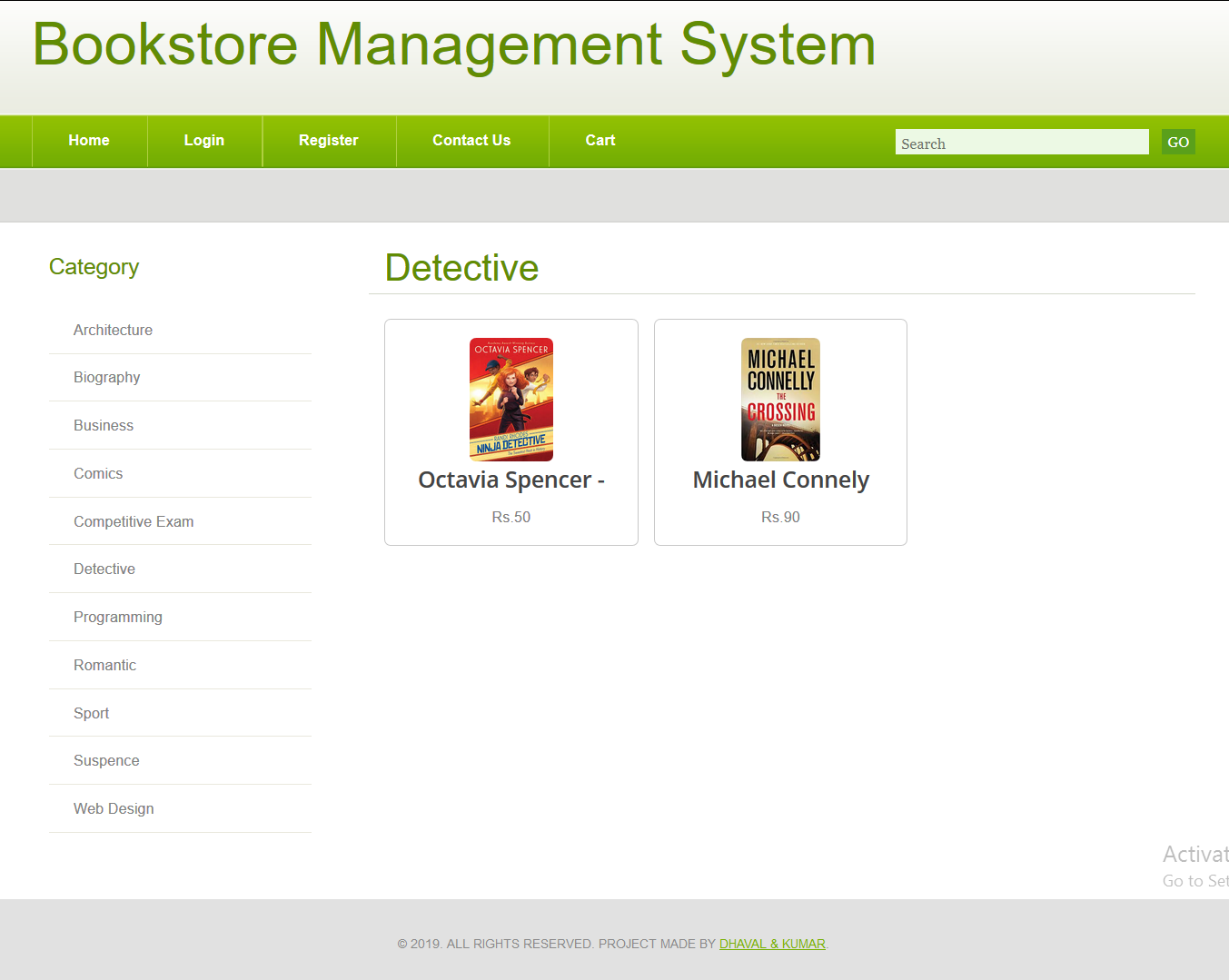
Book Category

### (Figure 16 : BMS Home Page)

1. **Selected Category**

Detective Category is selected.

Shows the Books of Detective Category.



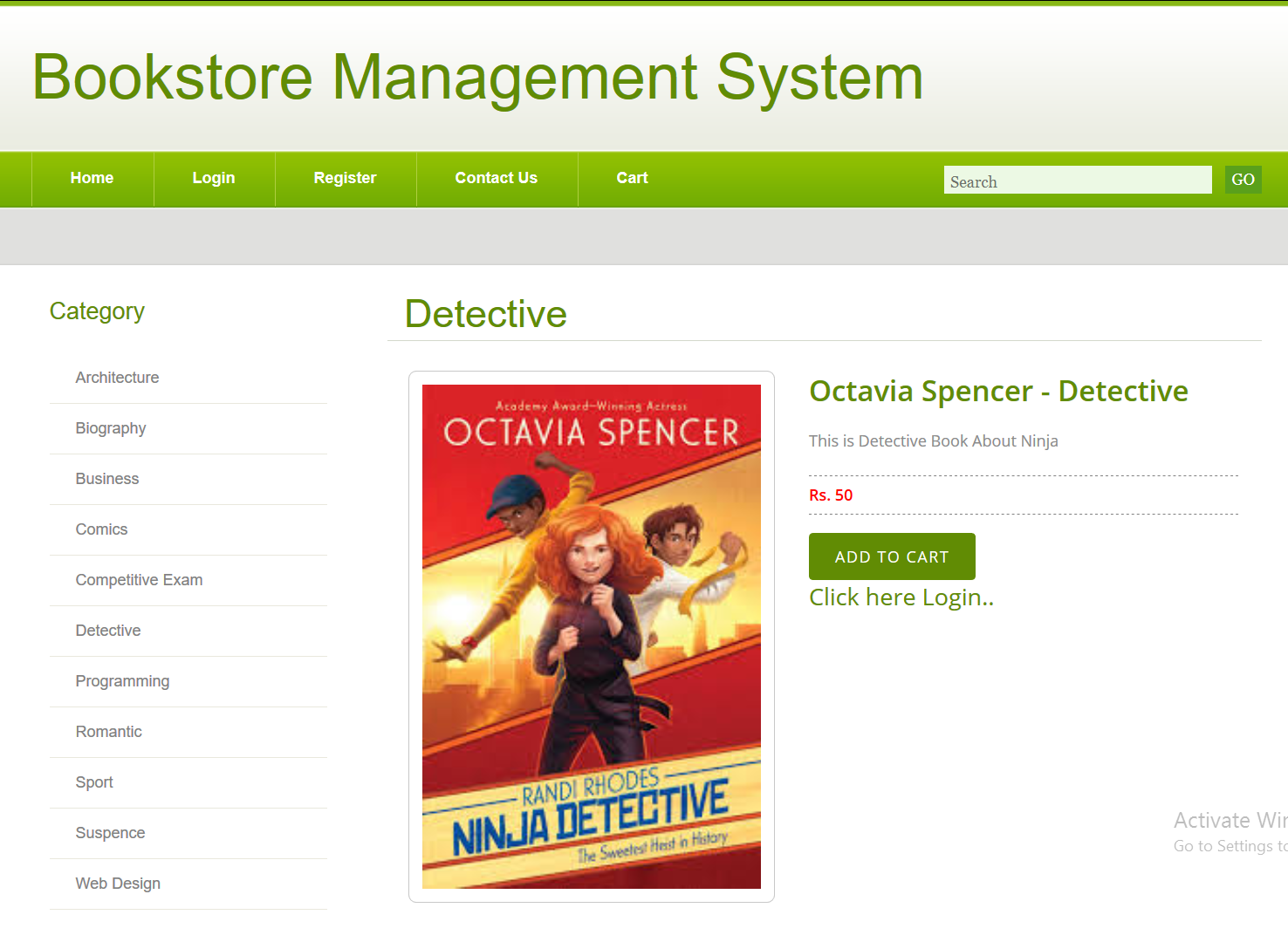
Books of Detective Category

### (Figure 17 : BMS Selected Category)

1. **Book Details (Before Login)**

Book Detail for Visitors.

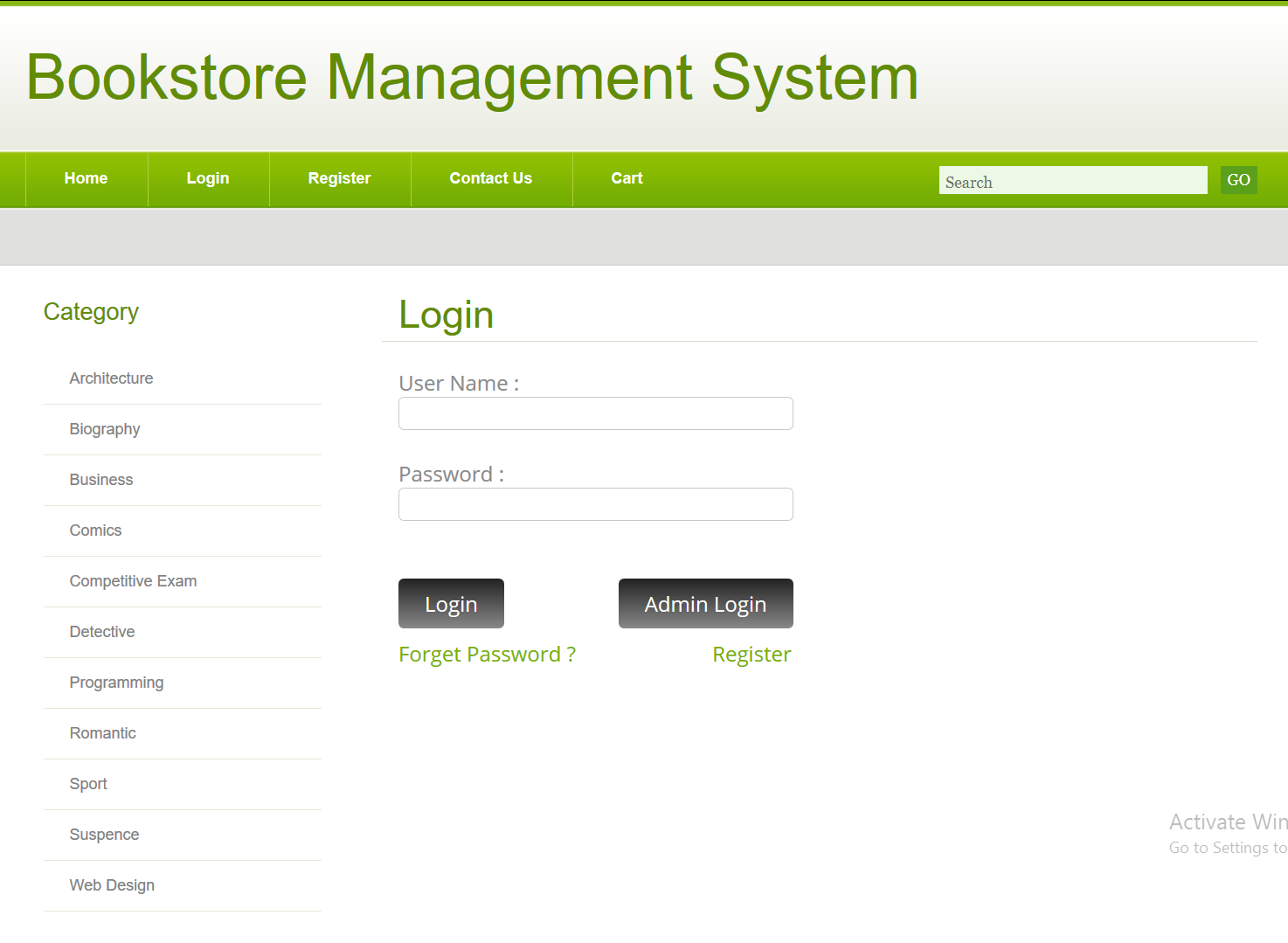
Visitors Can’t add Books to Add to Cart.



### (Figure 18 : BMS Book Details)

1. **Visitor Login Page**

Login Page for Viewers.



### (Figure 19 : BMS Login Page)

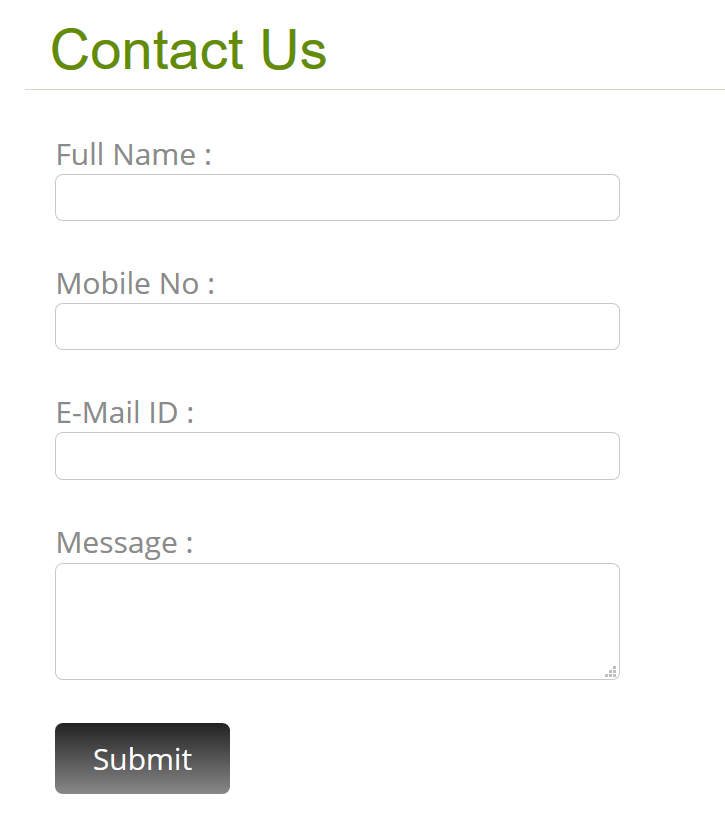
1. **Register Page**

Register Page for Viewers.



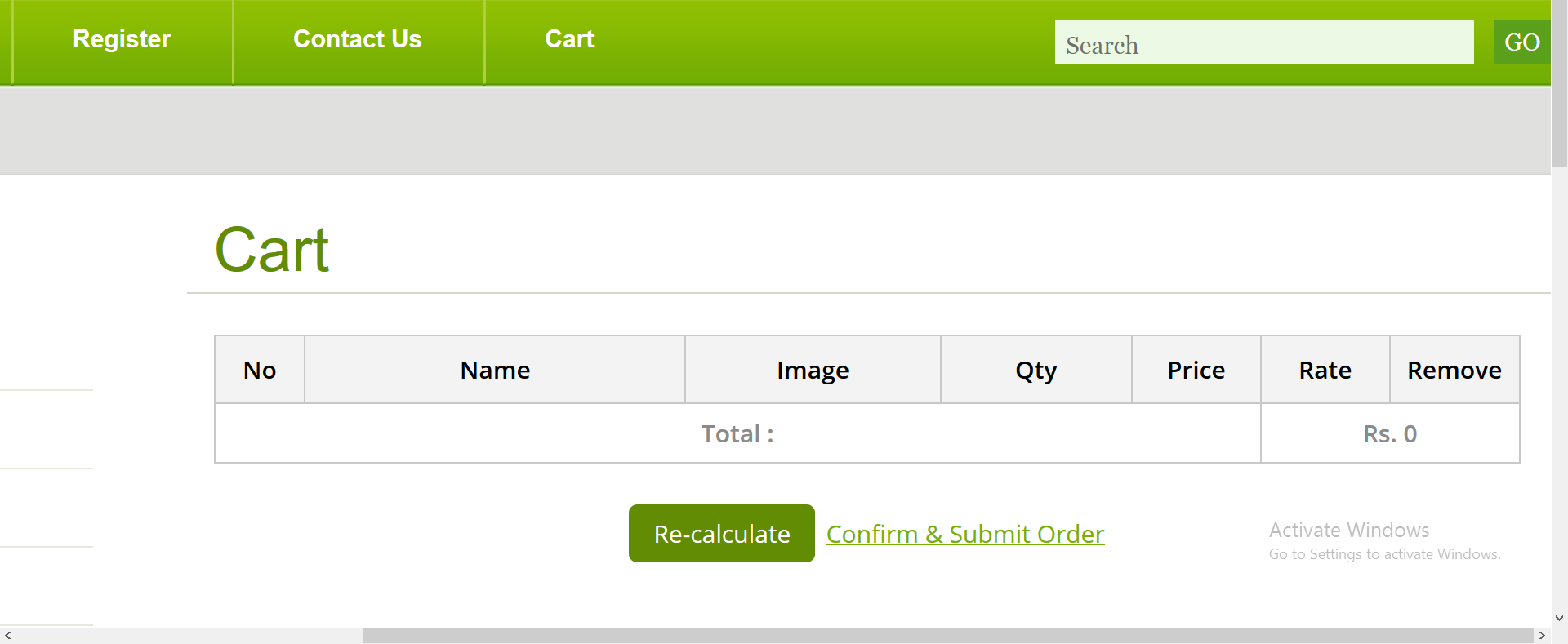
### (Figure 20 : BMS Register Page)

1. **Contact Us Page**



### (Figure 21 : BMS Contact Us Page)

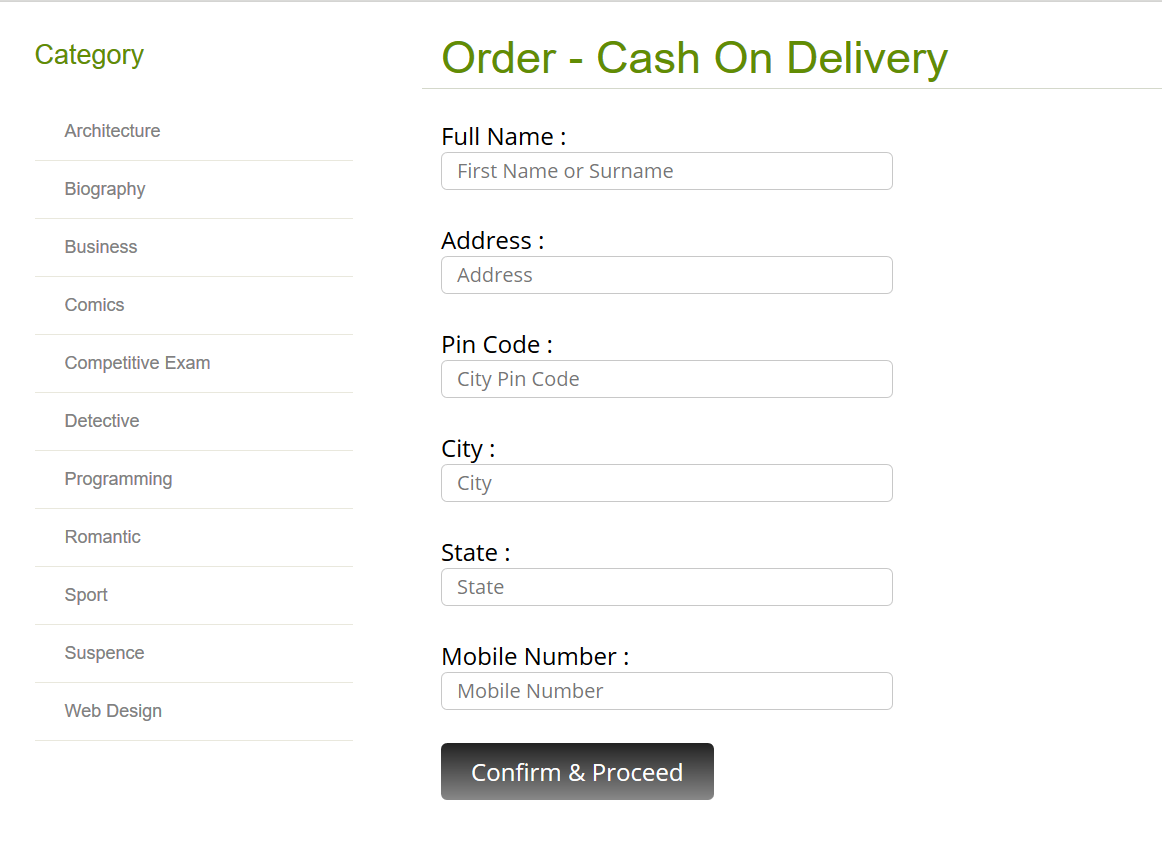
1. **Cart Page**



### (Figure 22 : BMS Cart Page Viewers)

1. **Order Page**

Only Cash On Delivery is Available for Orders.



### (Figure 23 : BMS Order Page)

1. **Home Page (Logged In)**

Automatically Navigation Bar Changed.

User Can Log Out.



### (Figure 24 : BMS Logged in Page)

1. **Book Details (Logged In)**

Users can Add Book To Add to cart.

Removed Sign in Link.



### (Figure 25 : BMS Users Book Details)

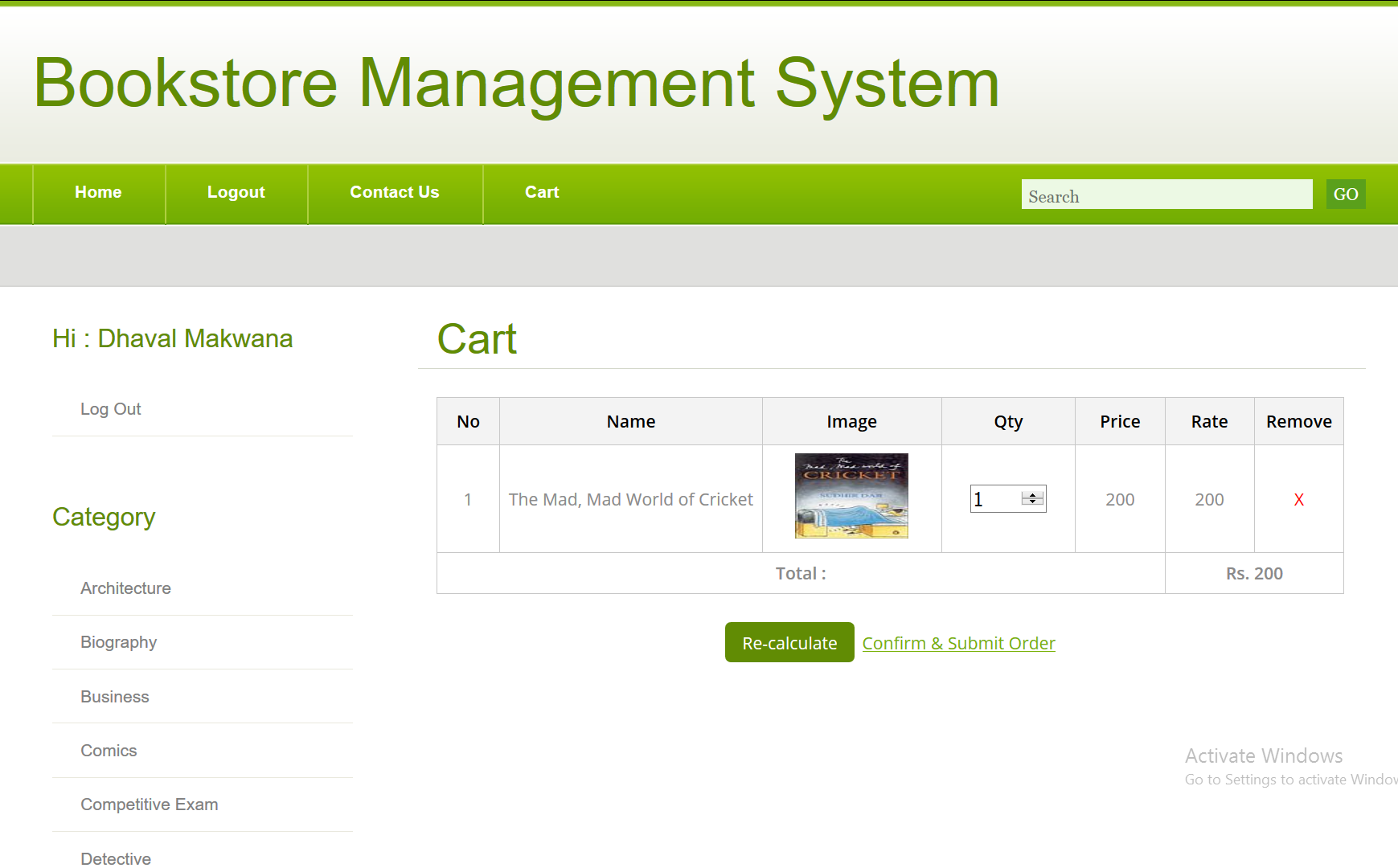
1. **Add to Cart (Logged In)**

Users Can add books to add to cart.

Details of books and price.

Click Recalculate to Qty, Rate and Total will Calculate.

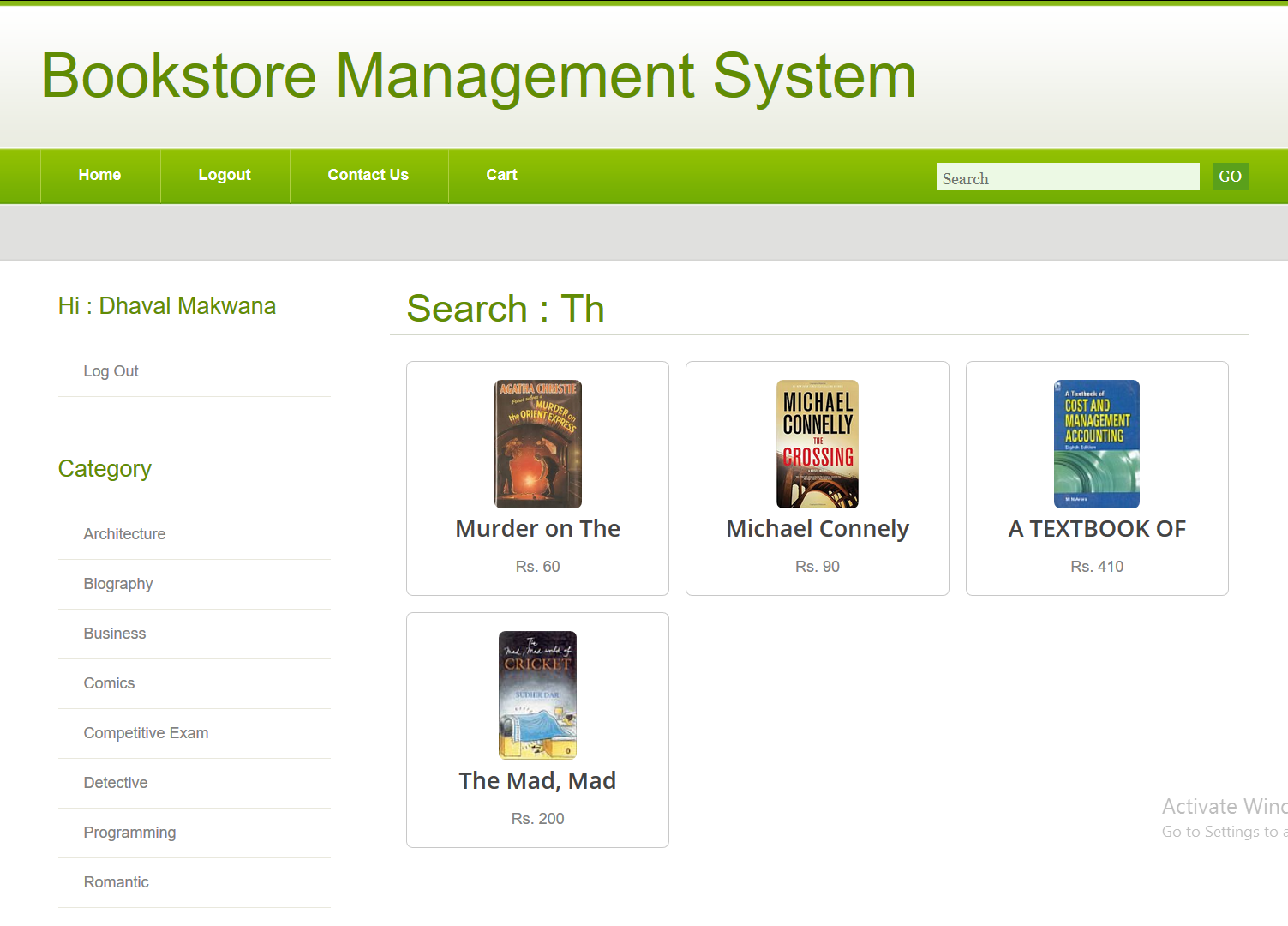
Users can order Books.



### (Figure 26 : BMS Users Cart Page)

1. **Search Books**

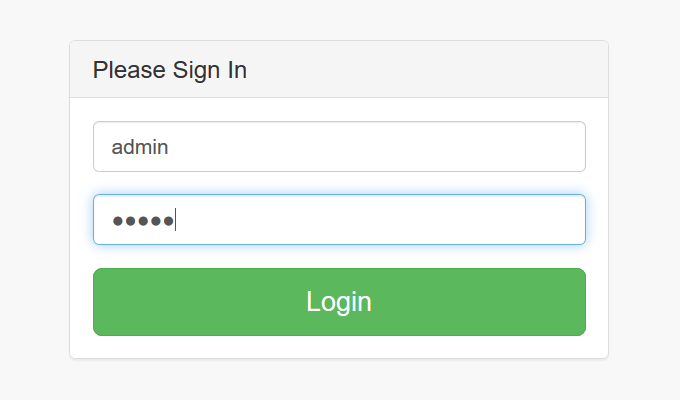
Book Search Feature.



Search Books

### (Figure 27 : BMS Search Books)

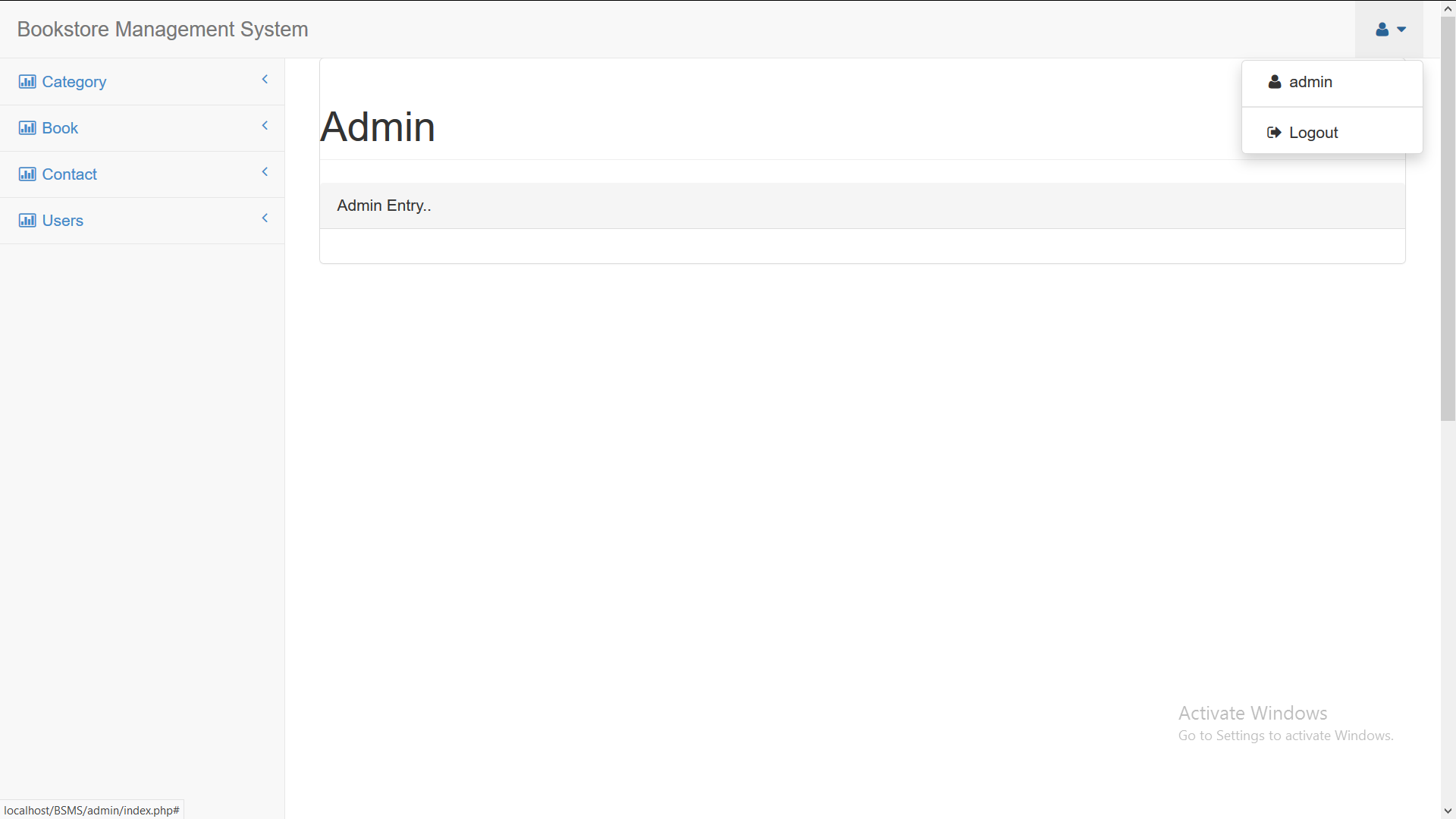
1. **Admin Login Page (New Template)**



### (Figure 28 : BMS Admin Login page)

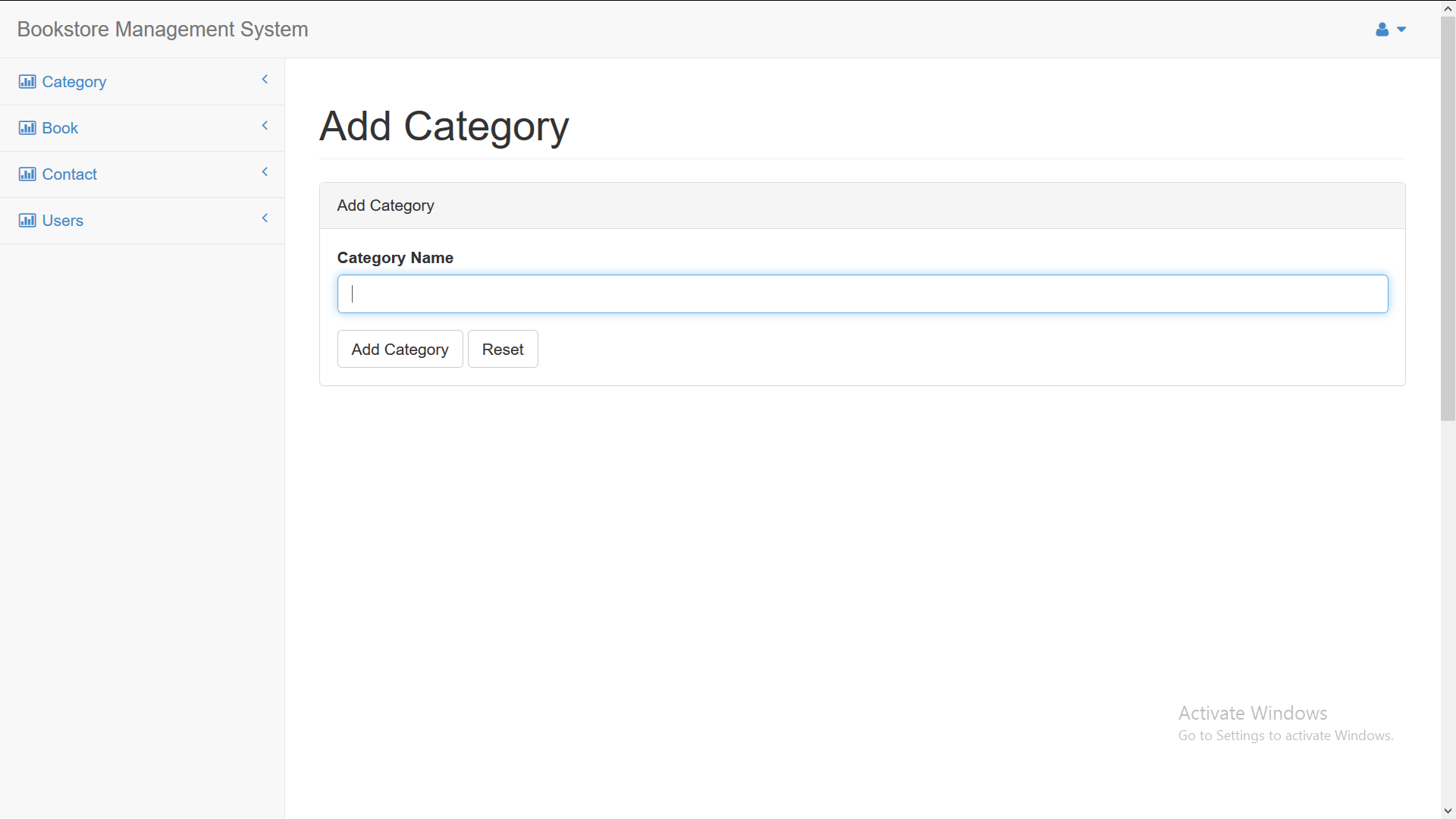
1. **Admin Home Page**

New Template.



### (Figure 29 : BMS Admin Home Page)

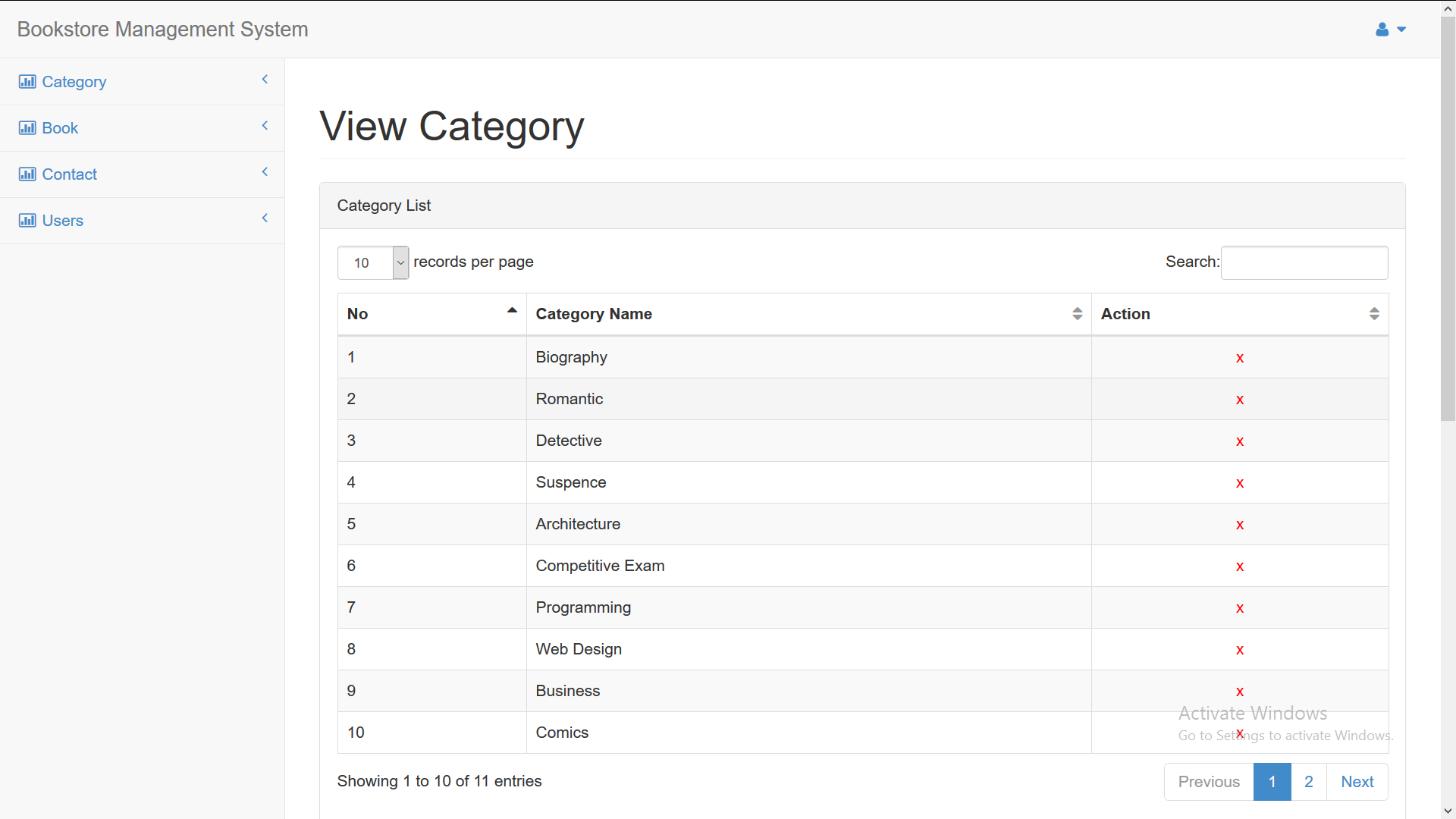
1. **Add Category (Admin)**



### (Figure 30 : BMS Add New Category)

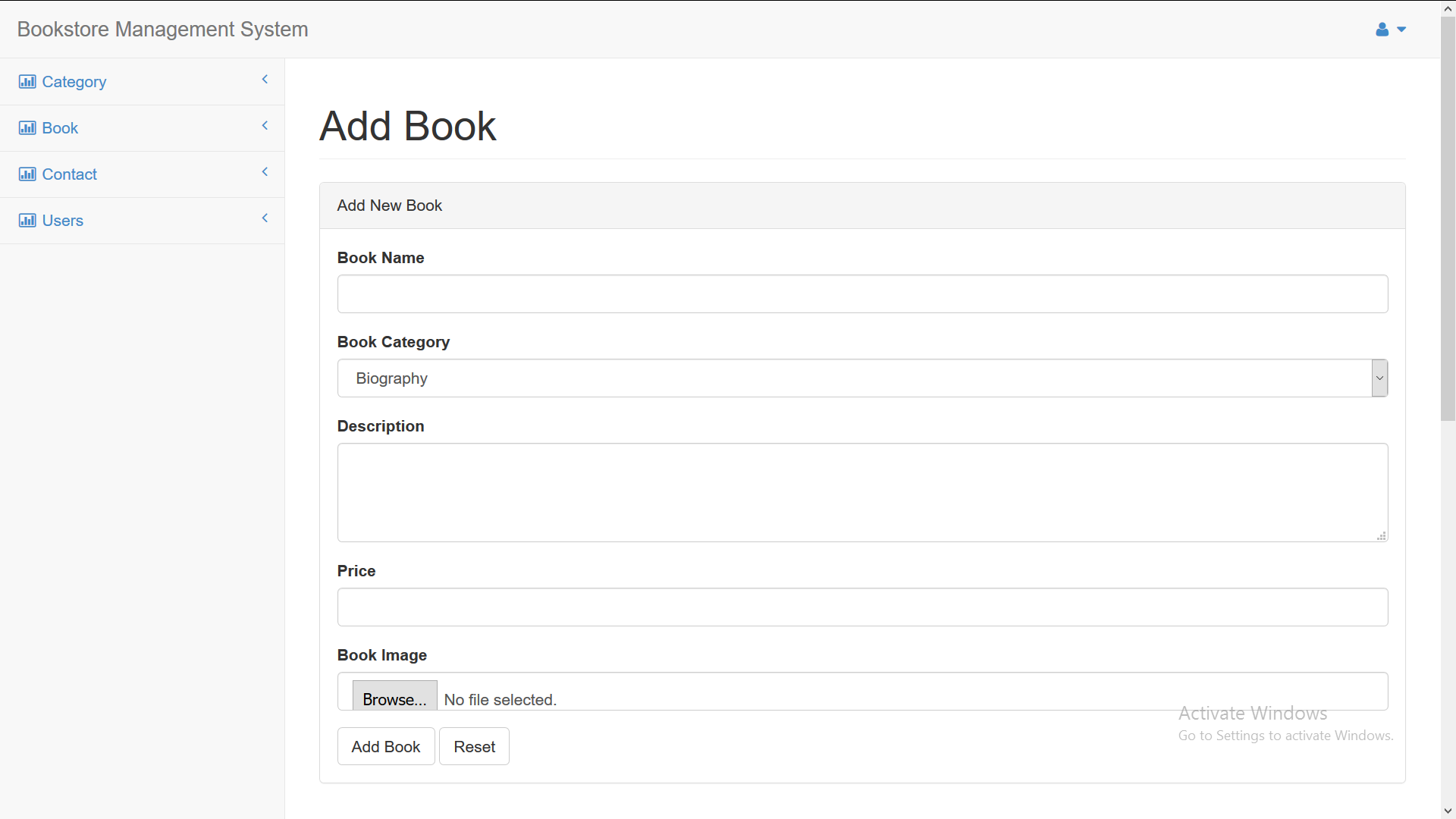
1. **View Category**

List of Books.

****

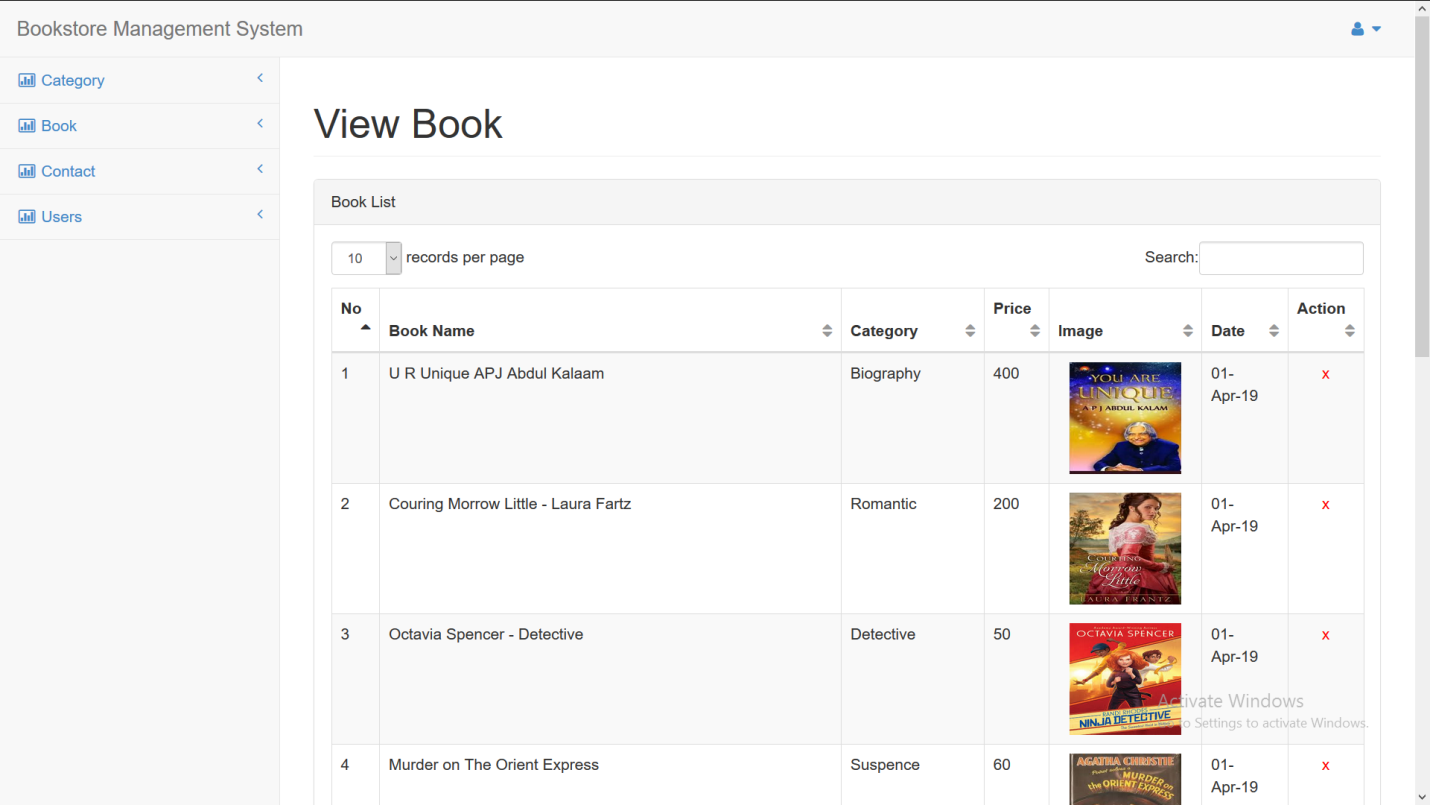
### (Figure 31 : BMS View Category)

1. **Add Books**

****(Figure 32 : BMS Add New Books)

#### View Books

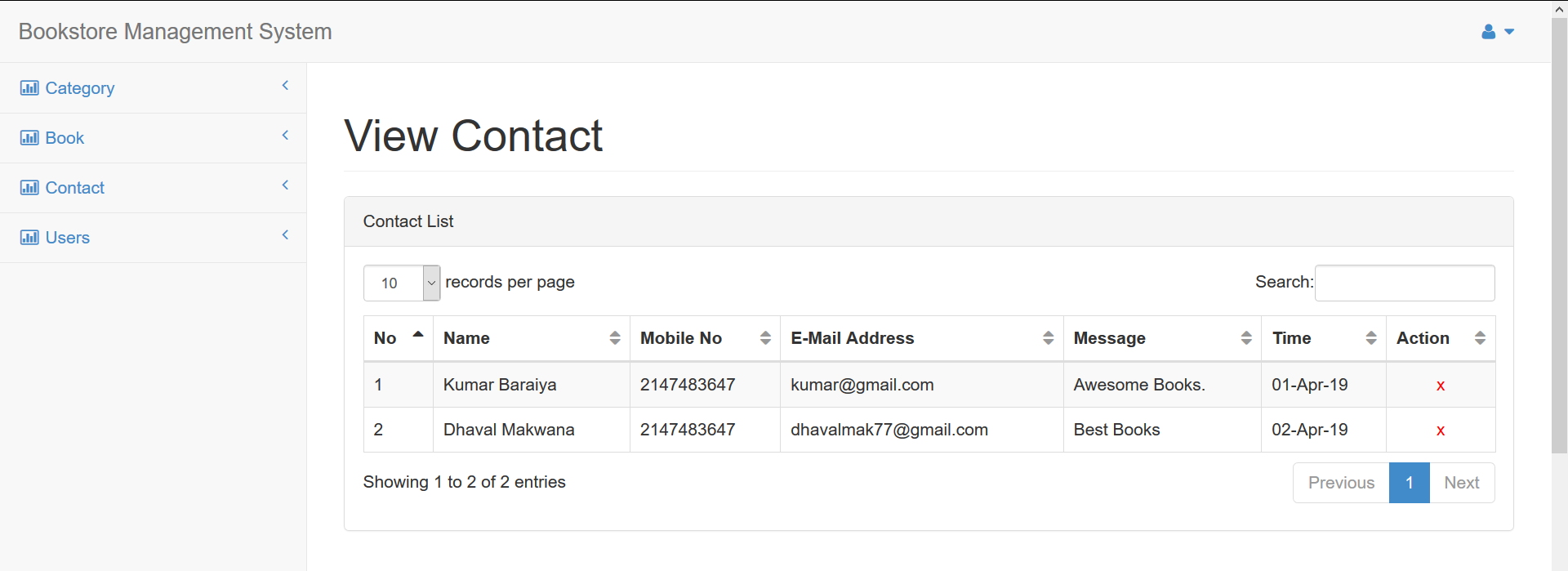
List Books for Admin.

****

### (Figure 33 : BMS View Books)

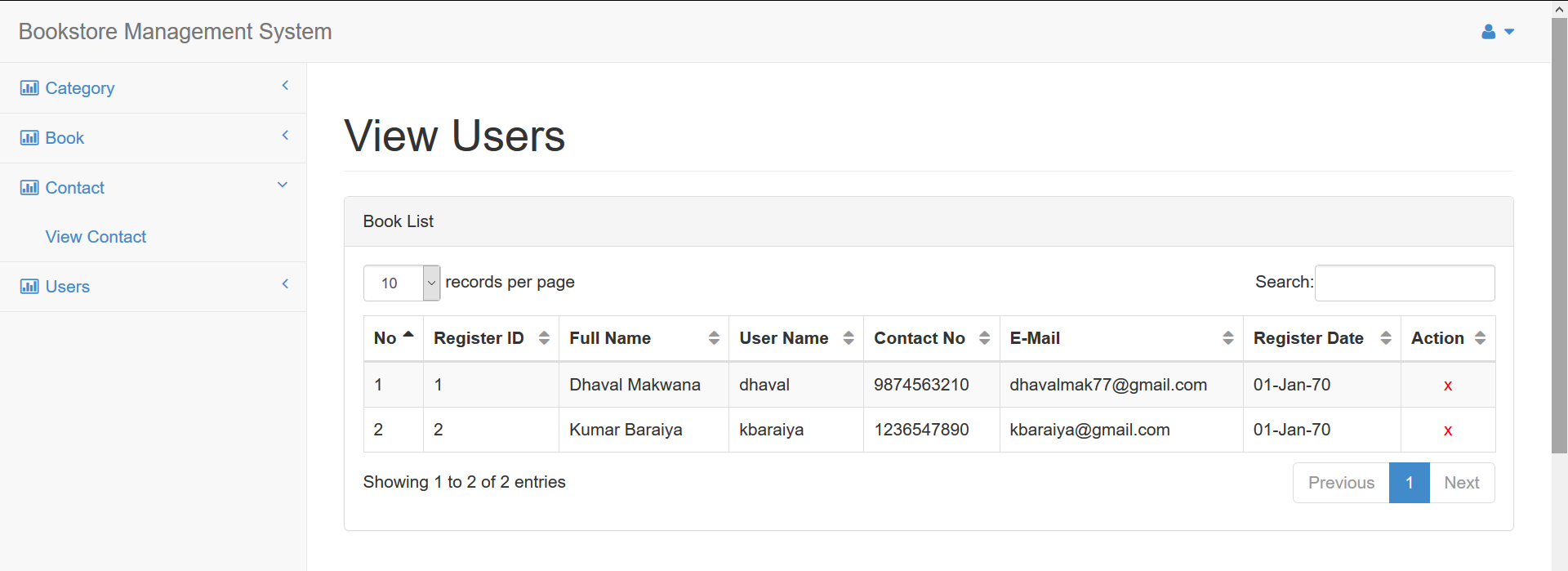
1. **View Contacted List**

List of People who Contacted using Contacted Page.

****

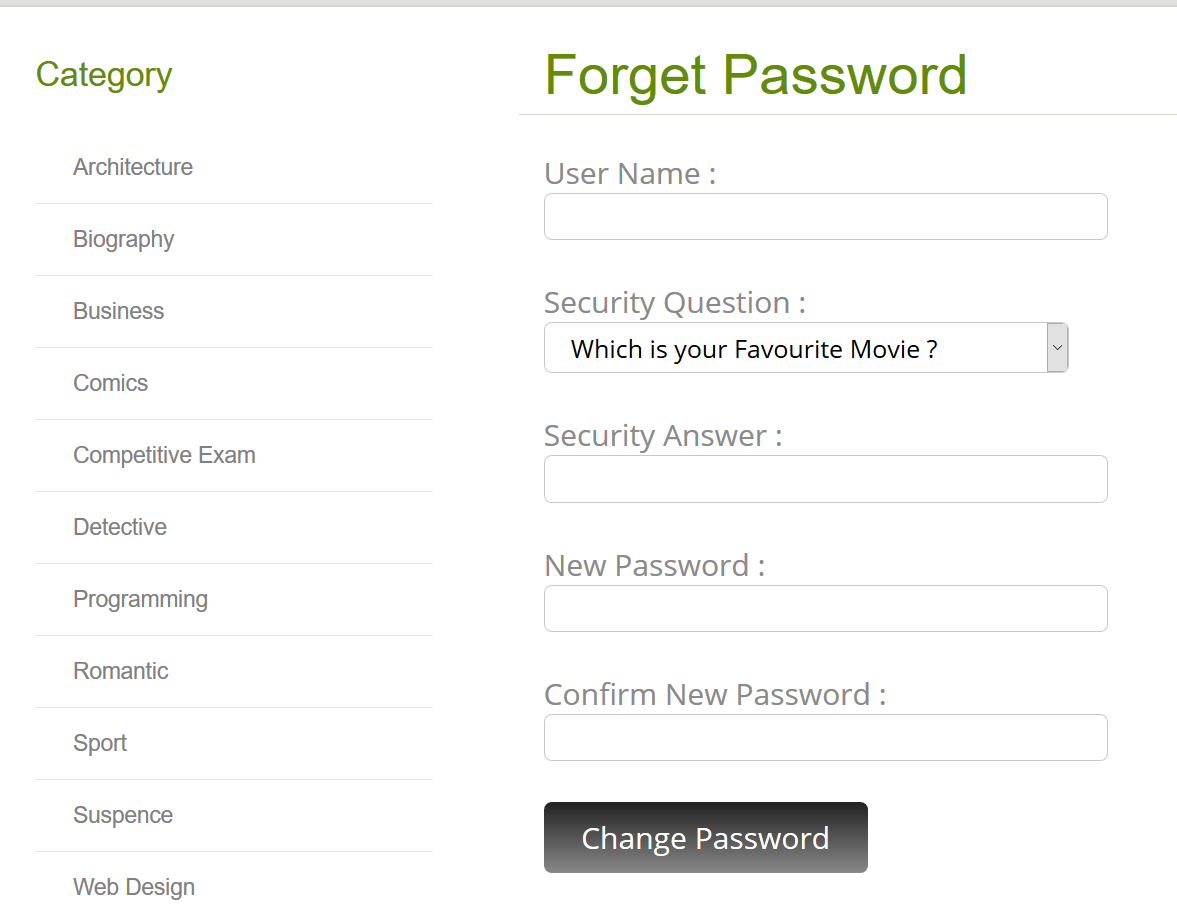
### (Figure 34 : BMS Contacted List Books)

1. **Users List**

****

### (Figure 35 : BMS Users List)

1. **Forget Password**

****(Figure 36 : BMS Forget Password Page)

# Chapter 4

# Testing And Implementation

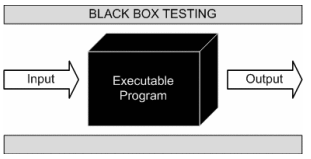
# 4.1 Testing Approach Used

* **Black box testing**

Black-box testing is a method of software testing that examines the functionality of an application based on the specifications. It is also known as specifications based .

Testing Independent testing team usually perform this type of testing during the software testing life cycle.

This method of test can be applied to each and every level of software testing such as unit, integration, system and acceptance testing**.**

****

### (Figure 37 : Black Box Testing)

This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories:

* + - Incorrect or missing functions
    - Interface errors
    - Errors in data structures or external database access
    - Behavior or performance errors
    - Initialization and termination errors

## Advantages of Black box Testing:

Tests are done from a user’s point of view and will help in exposing discrepancies in the specifications.

Tester need not know programming languages or how the software has been implemented.

* **White box testing**

White box testing is a testing technique That examines the program structure and derives test data from the program logic/code. The other names of glass box testing are clear box testing, open box testing, logic driven testing or path driven testing or structural testing.

White box testing involves looking at the structure of the code. When you know the internal structure of a product, tests can be conducted to ensure that the internal operations performed according to the specification. And all internal components have been adequately exercised.

* **White box testing techniques:**

1. Statement Coverage – This technique is aimed at exercising all programming statements with minimal tests.
2. Branch Coverage – This technique is running a series of tests to ensure that all branches are tested at least once.
3. Path coverage – This technique corresponds to testing all possible paths which means that each statement and branches is covered.

* **Advantages of white box Testing:**

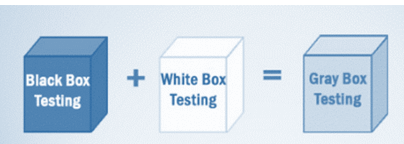
1. Forces test developer to reason carefully about implementation.
2. Reveals errors in “hidden” code.
3. Sports the code or other issues with respect to best programming practices.

* **Gray-box Testing:**

Grey-box testing is a testing technique performed with limited information about the internal functionality of the system. Grey-box testers have access to the detailed design information about requirements.

Grey box are generated based on the state based modes, UML diagrams or of the target system.

Grey Box Testing is a technique to test the software product or application with partial knowledge of the internal workings of an application.



### (Figure 38 : Gray Box Testing Testing)

# 4.2 Test Cases

**4.2.1 Admin Login Detail**

|  |  |  |  |
| --- | --- | --- | --- |
| Username | Admin | Password | Admin |
| Expected Result :   * If fields empty then gives a error for fill up fields * If password or username does not exist then gives error for valid detail. | | | |

**4.2.2 Login Detail**

|  |  |  |  |
| --- | --- | --- | --- |
| Username | Dhaval | Password | Dhaval |
| Expected Result :   * If fields empty then gives a error for fill up fields * If password or username does not exist then gives error for valid detail. | | | |

**4.2.3 Registration Details**

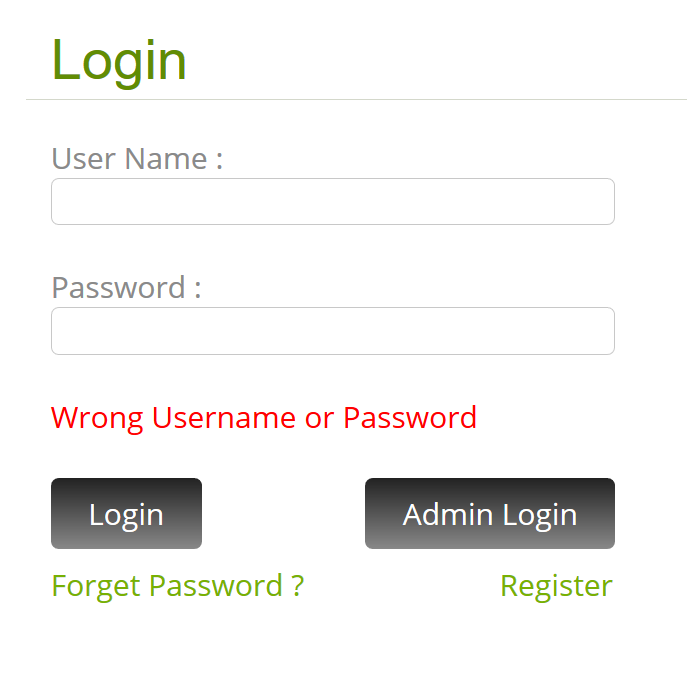
|  |  |  |  |
| --- | --- | --- | --- |
| Username | EMPTY | Password | EMPTY |
| Full Name | EMPTY | Security Answer | EMPTY |
| Expected Result :   * If fields empty then gives a error for fill up fields * If password or username does not exist then gives error for valid detail. * If password is < 8 characters then it will gives error. | | | |

**4.2.4 Order Details**

|  |  |  |  |
| --- | --- | --- | --- |
| Full Name | Address | Contact Number | EMPTY |
| Expected Result :   * If fields empty then gives a error for fill up fields * If contact number is not Numeric then gives error | | | |

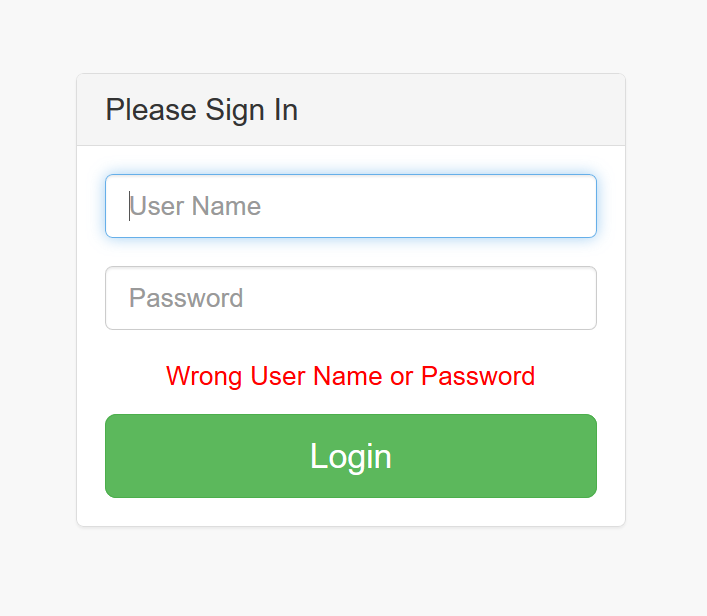
### Screen-Shots

1. **User Login**

****

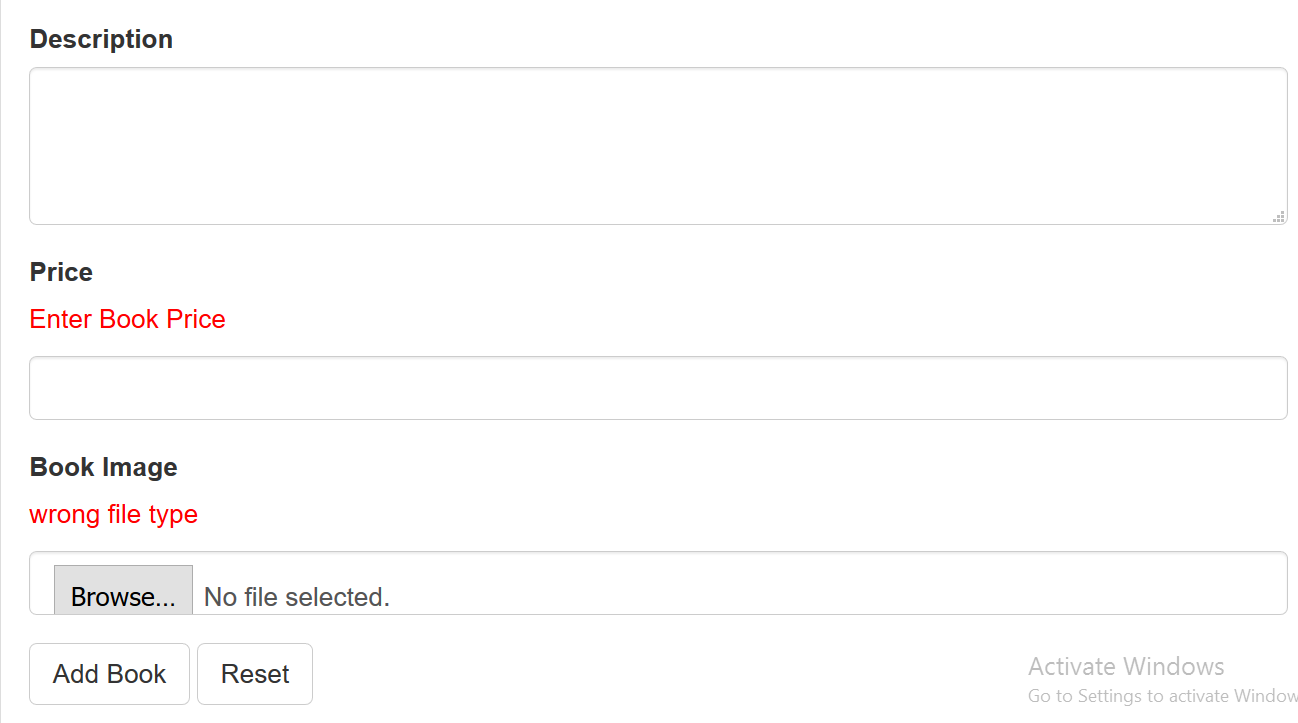
### (Figure 39 : Test Cases 1)

1. **Admin Login**

****

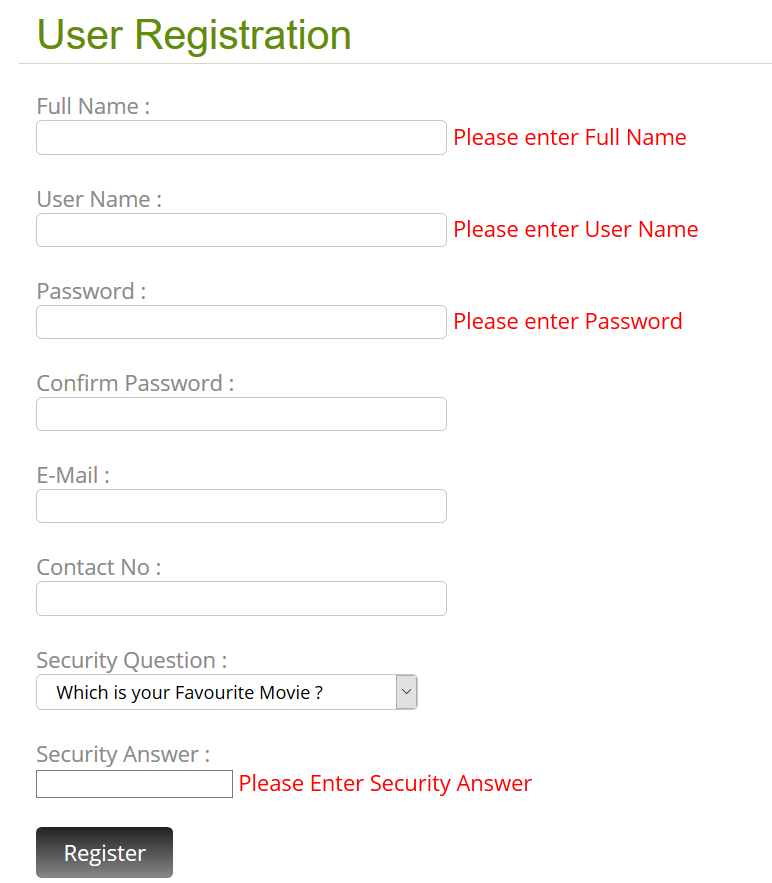
### (Figure 40 : Text Cases 2)

1. **Add Book**

****

### (Figure 41 : Text Cases 3)

1. **User Registration**

****

### (Figure 42 : Text Cases 4)

# Implementation approaches

Far the biggest challenge encountered was time constraints. Implementation takes an extraordinary amount of time and a large amount of coordination. Scheduling project meetings around every group member’s schedule has been nearly impossible. Many of the group members were unable to devote the amount of focus that the implementation stage required. Both the former and the latter problem may be more of an issue in the academic environment where priorities of the different group members are skewed in a variety of directions. Another issue that cropped up was knowledge of the PHP programming. At least two of the four group members were unfamiliar with PHP Swing API, which is php primary user interface package. Again, this may not be as much of an issue in software engineering outside the academic arena.

One of tools we found very useful, in situations where member responsibilities need to be hashed out, is the responsibility matrix. It has really been the only tool that has allowed us to continue making progress. Everyone is assigned a task, and everyone is held accountable for the completion of their assigned task. It also allows us to track tasks that need to be done. The responsibility matrix has proven to be an invaluable tool in the software engineering process.

# Chapter 5

# Conclusion

**Conclusion**

* At the first look we can say that Bookstore Management System is a perfect system but it has many limitations that are as follow :
* This is also used for list the category and books also manage the customer and books of the Bookstore.
* The Bookstore Management System is used to give information of the Books to the customer.
* We faced problems like Database creation, Flow of our system, designing of front end and back end tools, coding etc.
* Only single user can use a system at a time.
* In this system we cannot add a service module.
* We learnt new languages like jQuery, PHP, Boot-Strap, HTML, CSS, etc..

# 5.1 Limitation of system

* Help

Currently the help feature is not available. Using this functionality user can get help about the system.

* Payment

Currently the feature of online payment is not available. User cannot give payment online.

* Multilingual

Multilingual is not supported in our system. Therefore user cannot work in different languages.

* Backup & Recovery:

User cannot take the backup or recover the data in this the system.

* Many More Others.

# 5.2 Future Scope of the System

* Help module

Using this module user can get help on how to access the system. All functionalities of system are described in this module. And user can easily access the entire module using this feature.

* Online payment module

User can do their payment online using this functionality. In future we will add the online payment for make payment easier for the user.

* Multilingual

In this system we will add the multilingual therefore user can work in different languages and understand easily.

# 5.3 Bibliography

#### Websites Used

* + [www.google.com](http://www.google.com)
  + [www.w3cschools.com](http://www.w3cschools.com)
  + [www.stackoverflow.com](http://www.stackoverflow.com)
  + [www.quora.com](http://www.quora.com)
  + [www.Scribd.com](http://www.Scribd.com)

Apps Used

* Youtube
* Solo Learn
* Udemy..