**ONLINE BOOK SHOPPING**

**A Project Report**

Submitted in partial fulfillment of the

Requirements for the award of the Degree of B.Sc.I.T

**(BACHLOR OF SCIENCE IN INFORMATION TECHNOLOGY)**

**By T.Y.B.Sc.I.T**

Ritesh Eknath Tendolkar

Seat Number - 51

Vinod Dattatray Surve

Seat Number - 49

**UNDER THE GUIDANCE OF**

**Dr.Mohd.Ansari.Nasir**

**(Co-Ordinator Of Department Of Information Technology)**



**DEPARTMENT OF INFORMATION TECHNOLOGY**

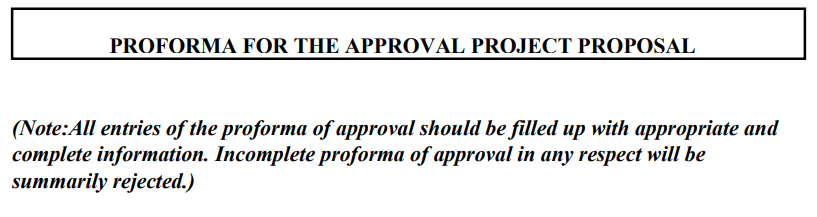
**K.P.B. HINDUJA COLLEGE OF COMMERCE**

***(Affiliated To University Of Mumbai)***

**MUMBAI 400004**

**MAHARASHTRA**

**2023-2024**

****

**PRN NO:** 2021016400939675  **ROLL NO:** 51

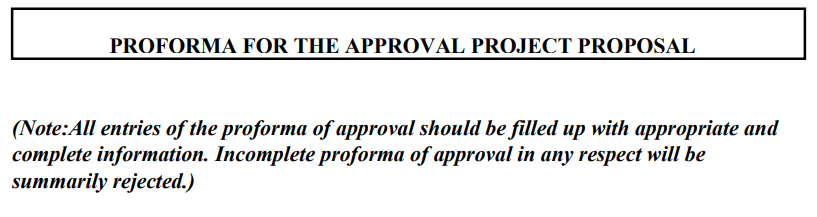
1. **Name of Student:** Tendolkar Ritesh Eknath
2. **Title of the Project:** Online Book Shopping
3. **Name of Guide:** DR. MOHD. ANSARI NASIR
4. **Teaching Experience of Guide:** 15 years
5. **Is this your first Submission:** Yes

**Signature of the Student Signature of the Guide**

**Date:** 2nd March 2024 **Date:** 2nd March 2024

**Signature of the Coordinator**

**Date:**

****

**PRN NO:** 2021016400927055 **ROLL NO:** 49

1. **Name of Student:** Surve Vinod Dattatray
2. **Title of the Project:** Online Book Shopping
3. **Name of Guide:** DR. MOHD. ANSARI NASIR
4. **Teaching Experience of Guide:** 15 years
5. **Is this your first Submission:** Yes

**Signature of the Student Signature of the Guide**

**Date:** 2nd March 2024 **Date:** 2nd March 2024

**Signature of the Coordinator**

**Date:**

**K.P.B. HINDUJA COLLEGE OF COMMERCE**

***(Affiliated to University of Mumbai)***

**MUMBAI-MAHARASHTRA-400004**

**DEPARTMENT OF INFORMATION TECHNOLOGY**



**CERTIFICATE**

This is to certify that the project entitled, "**Online Book Shopping**", is bonafied work of **Tendolkar Ritesh Eknath** bearing Seat.No: **1066730** submitted in partial fulfillment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from University of Mumbai.

**Coordinator Principal**

**Internal Examiner External Examiner**

**Date: College Seal**

**K.P.B. HINDUJA COLLEGE OF COMMERCE**

***(Affiliated to University of Mumbai)***

**MUMBAI-MAHARASHTRA-400004**

**DEPARTMENT OF INFORMATION TECHNOLOGY**



**CERTIFICATE**

This is to certify that the project entitled, "**Online Book Shopping**", is bonafied work of **Surve Vinod Dattatray** bearing Seat.No: **1066726** submitted in partial fulfillment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from University of Mumbai.

**Coordinator Principal**

**Internal Examiner External Examiner**

**Date: College Seal**

Smt. P.D. Hinduja Trust’s

K.P.B. Hinduja College of Commerce

(Autonomous)

NAAC Re-Accredited A+ (3rd Cycle )

College with Potential for Excellence by UGC

CERTIFICATE

This is to certify that Mr. / Miss / Mrs. **Tendolkar Ritesh Eknath** of **T. Y. B. Sc. [IT]**

**Roll No. 51 , University Exam Seat No. 2020426** 2023-24 has completed his/her practical work in the subject of **Software Project Management**, as required by the University of Mumbai for the partial fulfillment of **T. Y. B. Sc. IT SEM V** The information submitted is true and original to the best of my knowledge.

\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Co-Ordinator Dr. Ansari M. Nasir Dr. (Ms) M. B. Madlani

Principal

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Internal Examiner External Examiner College Seal

Smt. P.D. Hinduja Trust’s

K.P.B. Hinduja College of Commerce

(Autonomous)

NAAC Re-Accredited A+ (3rd Cycle )

College with Potential for Excellence by UGC

CERTIFICATE

This is to certify that Mr. / Miss / Mrs. **Surve Vinod Dattatray** of **T. Y. B. Sc. [IT]**

**Roll No. 49 , University Exam Seat No. 2020421** 2023-24 has completed his/her practical work in the subject of **Software Project Management**, as required by the University of Mumbai for the partial fulfillment of **T. Y. B. Sc. IT SEM V** The information submitted is true and original to the best of my knowledge.

\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Co-Ordinator Dr. Ansari M. Nasir Dr. (Ms) M. B. Madlani

Principal

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Internal Examiner External Examiner College Seal

**ABSTRACT**

The objective of this project is to develop an e-book store where books can be bought from the comfort of home through the Internet. An online book store is a virtual store on the Internet where customers can browse the catalog and select books of interest. The selected books may be collected in a shopping cart. The customer can then proceed to a checkout process, where the payment and delivery options are selected. The payment can be made using a credit card or other electronic payment methods. Once the payment is confirmed, the books are delivered to the customer's address. The system also provides an interface for the administrator to manage the books, customers, orders, and payments.

This project report on "Online Book Shopping" an e-commerce website that includes the following chapters:

- Introduction

- Survey of Technology

- Requirements and Analysis

- System Design

**ACKNOWLEDGEMENT**

We are pleased to present the “Online Book Shopping” project and take this opportunity to express our profound gratitude to all those who helped us complete this project.

We thank our college for providing us with excellent facilities that helped us to complete and present this project. We would also like to thank the staff members and lab assistants for permitting us to use computers in the lab as and when required.

We express our deepest gratitude towards our project guide for his/her valuable and timely advice during the various phases of our project. We would also like to thank him/her for providing us with all the proper facilities and support as the project co-coordinator. We would like to thank him/her for support, patience and faith in our capabilities and for giving us flexibility regarding working and reporting schedules.

We would like to thank all our friends for their smiles and friendship making the college life enjoyable and memorable. Family members always stood beside us and provided the utmost important moral support. Finally, we would like to thank everyone who has helped us directly or indirectly in our project.

**Table Of Contents**

**Chapter 1: Introduction**

* 1. Background
  2. Objectives
  3. Purpose, Scope, and Applicability
  4. Achievements

**Chapter 2: Survey of Technology**

* 1. Justification of selection of technology

**Chapter 3: Requirements and Analysis**

* 1. Problem Definition
  2. Working of Existing system
  3. Limitations of existing system
  4. Working of proposed system
  5. Features of proposed system
  6. Requirements Specification
     1. Hardware requirement
     2. Software requirement
  7. Requirement Analysis
     1. Fact Finding Techniques
        1. Observation
        2. Record review
        3. Interview
        4. Questionnaires
     2. Feasibility study
        1. Economical feasibility
        2. Technical feasibility
        3. Operational feasibility

**Chapter 4: System Design**

1. Basic Modules (Module division)
2. Schema/Database Design (Data Dictionary)
3. Gantt Chart
4. ER Diagram
5. Class Diagram
6. Object Diagram
7. Use case diagram
8. Collaboration/Sequence diagram
9. Activity diagram
10. State diagram
11. Package diagram
12. Component diagram
13. Deployment diagram

**Chapter 5: Implementation and Testing**

1. Code (Place Core segments) -Content includes description about coding phase in your project
2. Testing Approach (Black Box Testing and White Box Testing)
3. Unit Testing: Boundary Value Analysis (apply for project and write its test case with test result)
4. Equivalence Class Partitioning (apply for project and write its test case with test result)

**Chapter 6: Results** (Place Screen Shots and write the functionality of each screen at the bottom)

**Chapter 7: Conclusion and Future Work**

**Chapter 8: References** (Title of the book, Author, Full URL of online references etc.)

**Chapter 1**

**Introduction**

**Online shopping** is the process whereby consumers directly buy goods, services etc. from a seller interactively in real-time without an intermediary service over the internet.

Online shopping is the process of buying goods and services from merchants who sell on the Internet. Since the emergence of the World Wide Web, merchants have sought to sell their books to people who surf the Internet. Shoppers can visit web stores from the comfort of their homes and shop as they sit in front of the computer. Consumers buy a variety of items from online stores. In fact, people can purchase just about anything from companies that provide their books online. Books, clothing, household appliances, toys, hardware, software, and health insurance are just some of the hundreds of books consumers can buy from an online store.

Many people choose to conduct shopping online because of the convenience. For example, when a person shops at a brick-and-mortar store, she has to drive to the store, find a parking place, and walk throughout the store until she locates the books she needs. After finding the items she wants to purchase, she may often need to stand in long lines at the cash register.

* 1. **Background**

Online shopping allows you to browse through endless possibilities, and even offers merchandise that's unavailable in stores. If you're searching for a niche product that may not be distributed locally, you're sure to find what you're looking for on the internet. What's even more useful is the ability to compare items, similar or not, online. You can search through multiple stores at the same time, comparing material quality, sizes and pricing simultaneously.

Online bookstores offer a wide range of books, including textbooks, novels, biographies, and more. They also provide customers with the ability to search for books by author, title, or subject matter.

Say 'goodbye' to the days when you stood in line waiting, and waiting, and waiting some more for a store clerk to finally check out your items. Online shopping transactions occur instantly-saving you time to get your other errands done! Additionally, unlike a store, online shopping has friendly customer service representatives available 24 hours a day, 7 days a week to assist you with locating, purchasing and shipping your merchandise.

* 1. **Objective**

Our objective is to design such an application using which one can say 'goodbye' to the days when you stood in line waiting, and waiting some more for a store clerk to finally check out your items. The objective of an online book shopping (e-commerce) website is to provide a platform where customers can browse, select, and purchase books from the comfort of their homes. Here are some specific objectives –

* Online bookstores are accessible 24/7, allowing customers to shop at their convenience.
* They offer a vast selection of books across various genres, authors, and languages.
* They provide efficient search and filter options to help customers find specific books.
* They offer home delivery services, sometimes even globally.

My main aim is to design such a book store where customer can visit our site anytime of the day from anywhere to view the available books, choose any of them and can order by paying online or can opt for cash on delivery as well. The administrator will regularly add any new books available to them for sale. The administrator will take books from the reputed publishers and vendors only.

* 1. **Purpose and Scope**
     1. **Purpose**

The purpose of an online book shopping e-commerce website is to provide customers with a convenient and affordable way to purchase books. Online book shopping offers a number of advantages over traditional brick-and-mortar bookstores, including:

* Convenience: Customers can shop for books online from anywhere in the world, at any time of day or night. They do not need to travel to a physical store or wait in line to purchase a book.
* Selection: Online book stores typically have a much wider selection of books than brick-and-mortar stores. This is because online bookstores do not have the same physical space constraints as brick-and-mortar stores.
* Price: Online book stores often offer lower prices than brick-and-mortar stores. This is because online book stores have lower overhead costs.

Reviews: Online bookstores typically allow customers to read and write reviews of books. This can help customers make informed decisions about which books to purchase.

* + 1. **Scope**

In project management, the “scope” refers to the specific goals, deliverables, features, and tasks that define the boundaries of the project.

Complete E-Commerce Functionality: Develop a comprehensive e-commerce platform that allows users to browse products, add items to cart, and make purchases. This includes:

* Product Listing: Display a wide range of books available for purchase.
* Shopping Cart: Allow users to add their chosen books to a shopping cart.
* Checkout System: Enable users to review their cart and proceed to checkout.

User and Admin Dashboard: Implement a user-friendly interface for both customers and administrators. This includes:

* User Dashboard: Provide customers with a personal dashboard where they can view their purchase history, track their orders, and manage their account details.
* Admin Dashboard: Develop an admin dashboard for managing product listings, tracking sales, and handling customer inquiries.

. Advanced Features: Incorporate advanced features to enhance user experience and streamline operations. This includes:

* Payment Options: Integrate multiple payment options (credit/debit cards, net banking, digital wallets) to provide flexibility to customers.
* Responsive Design: Ensure the website is responsive and provides optimal viewing experience across a range of devices (desktop, mobile, tablet).
* Email Notifications: Implement an automated system for sending emails to users regarding order confirmation, shipping details, and other updates.
* Online Deployment: Deploy the project online to make it accessible to users across the globe.

The main issues being addressed in the project are:

* The project will address the need for advanced features such as diverse payment options, responsive web design.
* Protecting customer data and ensuring secure transactions are critical issues that the project will address.
* SQL injection attacks pose a significant threat to database security by manipulating SQL queries through user inputs.
  1. **Achievements**

Through this project, we achieved a deep understanding of e-commerce principles and website development. We had successfully designed and implemented an online book shopping platform that meets the objectives of providing convenience, a wide selection of books, efficient search options, and user-friendly interfaces for both customers and administrators.

Through the development process, we gained hands-on experience in various technical aspects such as website development, database management, user interface design, and payment integration, thus expanding our technical skill set.

The project emphasized user-centric design principles, resulting in an intuitive and user-friendly interface that enhances the overall shopping experience for customers, fostering increased user engagement and satisfaction.

**Chapter 2**

**Survey of Technology**

**Latest technology**

**JavaScript :-** It is a lightweight, interpreted, object-oriented programming language that is mainly used for client-side web development. It enables dynamic interactivity on web pages by manipulating the HTML document object model (DOM) and applying various effects and features. It also supports scripting, first-class functions, prototype-based inheritance, and asynchronous and event-driven programming.

**Java :-** It is a **general-purpose, object-oriented, platform-independent** programming language that is widely used for creating **web, desktop, and mobile applications**. It also supports **robust, secure, and high-performance** features, such as automatic garbage management, integrated development environment (IDE), and a rich application programming interface (API). Java has a large and active community of developers and good documentation.

**Python** **:-** It is a **high-level, interpreted, multi-paradigm** programming language that is widely used for creating **web, desktop, and data science applications**. It also supports **simple, readable, and expressive** features, such as dynamic typing, multiple inheritance, indentation-based syntax, and built-in data structures. Python has a large and diverse community of developers and a rich collection of libraries and frameworks.

**Crystal** **:-** It is a **compiled, object-oriented, systems programming language** that aims to combine the **speed and efficiency of C** with the **readability and friendliness of Ruby**. Crystal has a syntax close to Ruby’s and features **statically inferred types, C bindings, and macros.** Crystal can be used for creating web, desktop, and blockchain applications.

**Dart :-** It is an **open-source, object-oriented, client-optimized programming language** that is widely used to develop **mobile, web, desktop, and IoT applications** using the Flutter framework. Dart has a C-style syntax and supports **advanced concepts** such as interfaces, mixins, abstract classes, generics, and null safety. Dart also supports **asynchronous programming** with futures and streams.

**Raspbian :-** It is a **free operating system based on Debian** optimized for the **Raspberry Pi** hardware. Raspberry Pi is a **mini-computer** that can be used for various projects, such as robotics, gaming, or smart home. Raspbian comes with over **35,000 packages** of pre-compiled software and utilities that make it easy to install and use on your Raspberry Pi. Raspbian also has some **new features** added in 2018, such as a setup wizard, a recommended software tool, a new PDF viewer, and network booting.

**Windows :-** It is a **popular operating system** developed by Microsoft that runs on personal computers, laptops, tablets, and other devices. Windows has many **features and components** that allow users to perform various tasks, such as browsing the web, managing files, playing games, and working with applications. Windows also lets users **customize** their settings, appearance, and preferences. Windows has different **versions** and **editions** that cater to different needs and preferences of users.

**MacOS :-** Itis an **operating system** developed by Apple that runs on Mac computers. macOS is **graphically based**, meaning that users can interact with icons, menus, and windows using a mouse or a trackpad. macOS has many **features and tools** that allow users to do various tasks, such as browsing the web, checking email, editing photos, listening to music, and playing games. macOS also supports **Handoff** and **Universal Clipboard**, which let users seamlessly switch between Mac and other Apple devices.

**Ubuntu :-** It is a **free and open-source operating system** based on Linux that runs on desktops, laptops, servers, and cloud platforms. Ubuntu is **easy to install, manage, and use**, and supports a wide range of hardware and software. Ubuntu has a **simple and intuitive interface** that lets users launch applications from a launcher or a dash. Ubuntu also has thousands of **apps available for download** from its software center, including office, web, email, photo, and gaming apps.

**Solaris :-** It is an **enterprise-class operating system** developed by Oracle that runs on SPARC and x86-64 systems. Solaris is known for its **scalability, security, and innovation**, and has many features such as **DTrace, ZFS, and Solaris Containers.** Solaris can be installed from various software groups, depending on the user’s needs. Solaris also supports **software-defined networking** and **built-in virtualization** for deploying cloud applications.

**Acros :-** It is a **modern operating system** based on OS/2 that is developed by Arca Noae. Acros supports **symmetric multiprocessing systems** with up to 64 processor cores, and is **ACPI 6.1-compliant**. Acros also has **hardware compatibility** with various devices, such as USB, network, audio, and video. Acros is designed for **business and personal use**, and can run many OS/2, DOS, and Windows applications.

**Oracle :-** It is a **relational database management system** developed by Oracle Corporation. Oracle is used to store, manage, and manipulate data for various applications and purposes. Oracle has many **features** that make it powerful, scalable, and reliable, such as **Real Application Clustering, Portability, Diagnosability, and Performance**. Oracle also has different **versions and editions** that offer different functionalities and options for users. Oracle is widely used by businesses and organizations for its **advantages** such as portability, security, and compatibility.

**MySQL** **:-** It is a **relational database management system (RDBMS)** that uses SQL (Structured Query Language) to store, manage, and manipulate data. MySQL is **open-source and free** under the GNU license, and supported by Oracle Corporation. MySQL has many **features and benefits** that make it popular and widely used, such as **ease of use, security, scalability, reliability, and performance**. MySQL can be used for various applications and purposes, such as data warehousing, online shopping, logging software, and portals.

**MongoDB** **:-** It is a **NoSQL document database** that stores data in flexible JSON-like documents. MongoDB is **schema-less**, meaning that documents in the same collection can have different structures and fields. MongoDB has many **features** that make it powerful, scalable, and easy to use, such as **sharding, replication, indexing, and ad hoc queries**. MongoDB can be used for various applications and purposes, such as big data analytics, content management, real-time applications, and IoT.

**Language Used In Project**

**HTML** (HyperText Markup Language) is a **standard language** for creating web pages and web applications. HTML can be used in a Online Book Shopping website to **display and format** the content of the web pages, such as the Categories and Sub-Categories, the book list, the login form, the search function, etc. HTML can also be used to **link** to other web pages or resources, such as images, CSS, JavaScript, PHP, or MySQL. HTML is essential for creating a **user-friendly and interactive** library website.

**JavaScript** is a **scripting language** that can run in the web browser and on the server. JavaScript can be used in a Online Book Shopping website to **add functionality and interactivity** to the web pages, such as validating the login form, searching for books, etc. JavaScript can also be used to **communicate** with the backend server or database, such as using AJAX, Node.js, Express, or MongoDB. JavaScript is useful for creating a **dynamic and responsive** website.

**CSS** Like HTML, CSS is not a programming language. It's not a markup language either. CSS is a style sheet language. CSS is what you use to selectively style HTML elements. CSS is a language that is used to style and layout web pages. It can change the appearance of elements such as fonts, colors, backgrounds, and more. It can also create animations and other decorative features. CSS works by applying rules to elements that match certain selectors. CSS can be written in external files, internal style blocks, or inline attributes.

**Bootstrap** is a **popular and free CSS framework** that provides ready-made components and templates for creating web pages and web applications. Bootstrap can be used in a Online Book Shopping website to **save time and effort** in designing and developing the web pages, such as the navbar, the form, etc. Bootstrap can also be used to **ensure responsiveness and compatibility** across different browsers and devices. Bootstrap is useful for creating a **professional and consistent** website.

**PHP, or Hypertext Preprocessor** is a server-side scripting language used in web development. In an online book shopping website, PHP is used for server-side processing, database interaction, generating dynamic content, user authentication, and handling backend logic. It plays a crucial role in the functionality and operation of the website.

**VSCode** is a tool that can make and change code for websites and apps. You can use VSCode to change website files from anywhere using FTP or GitHub or Azure Repos. You can also use VSCode to test and fix your code, add more features, and change your settings.

Google Chrome, as a web browser, plays a crucial role in online book shopping. It serves as the platform where users interact with the online bookstore. Users can search for books, read descriptions, add books to their cart, and make purchases all within the Chrome browser.

**Chapter 3**

**Requirement and Analysis**

* 1. **Problem Definition**

**Current online book shopping websites often face challenges with: -**

Limited search options and filters: Difficulty finding specific books based on detailed criteria

Inefficient inventory management: Stock levels are inaccurate, leading to unexpected delays or cancellations.

Unintuitive user interface: Navigation is difficult and checkout process is complex.

Limited customer support options: Difficulty contacting customer service for assistance.

Need for New system:-

* Online Book Store is a specific requirement of the client that integrates the buying and selling services specifically to their customers.
* Reports can be generated at any time within few seconds, so that manual labor is not required, and also analysis can be performed much more frequently which helps in taking decision.
* The details regarding all users, books can also be maintained as their information is very helpful and sometimes becomes a critical requirement.
* Allows user to get registered from their places and transact for the required product.
* To overcome these problems we develop “Online Book Store”.
* Offer advanced search and filtering features: Allow users to easily find specific books by genre, author, publication date, rating, format, etc.
* Utilize accurate inventory management systems: Ensure real-time stock information to avoid delays and cancellations.
* Provide a user-friendly interface: Make navigation and checkout process simple and intuitive.
* Offer various customer support options: Provide live chat, email, phone support, and FAQs to assist customers.

By implementing these improvements, the new system can:

* Increase customer satisfaction and retention.
* Improve sales and revenue.
* Enhance brand image and reputation.
* Gain a competitive advantage in the online book market.
  1. **Working of Existing system**

A traditional bookstore is a physical retail location where people can buy books in person. Here's a simple explanation of how it works:

1. Setting Up: The bookstore typically has a storefront or a building where books are displayed and sold. It may have shelves, racks, or tables to showcase various titles.
2. Inventory: The store stocks a variety of books across different genres and categories. These books may be new releases, bestsellers, classics, or niche titles.
3. . Browsing: Customers enter the store and browse through the selection of books. They can pick up books of interest, read blurbs, flip through pages, and decide whether they want to purchase them.
4. Purchase: When a customer decides to buy a book, they take it to the cashier or checkout counter. The cashier scans the book's barcode or manually enters the book's details into the system.
5. Payment: The customer then pays for the book using cash, credit/debit card, or other accepted payment methods.
6. . Returns and Exchanges\*: If a customer needs to return or exchange a book, they can usually do so within a specified time frame, provided the book is in resalable condition.
7. Staff Assistance\*: Bookstore staff members are available to assist customers in finding books, providing recommendations, answering questions, and offering assistance as needed.

Overall, traditional bookstores provide a physical space for people to explore, discover, and purchase books while also offering a unique social and cultural experience.

* 1. **Limitations of existing system**

Traditional bookstores have some limitations:

1. Limited Selection: Traditional bookstores can only carry a certain number of books due to physical space constraints. This means they might not have the book you're looking for.
2. Location Dependency: You have to physically go to the bookstore, which might not be convenient if it's far away or if you don't have transportation.
3. Operating Hours: Bookstores have specific opening and closing times, so you can't shop whenever you want.
4. No Instant Access to Information: If you have a question about a book or need more information, you might have to rely solely on the knowledge of the staff, which could be limited.
5. Limited Inventory: Physical space constraints mean that traditional bookstores can only stock a finite number of titles. This limitation may result in a narrower selection compared to online retailers or larger bookstores.
6. Geographical Constraints: Customers must physically visit the store to make purchases, which can be inconvenient for those living far away or without easy access to transportation.
7. Operating Costs: Maintaining a brick-and-mortar location entails expenses such as rent, utilities, and staffing. These overhead costs can make it challenging for traditional bookstores to compete with online retailers that have lower operating expenses.
   1. **Working of proposed system**

Website comprises two modules with their sub modules as follows:

* **Admin**
* Login:
* The admin can log in using their credentials.
* Insert and View Book:
  + It displays a list of all the books available on the website, providing basic information such as the book title, author and also admin can add books.
  + It also displays entries .
* Order list
* It provides a comprehensive overview of all the orders placed by users.
* Each order record includes essential details such as invoice number, order date, total price and order status.
* Category and Sub-Category
  + The admin can add categories and its sub-categories as well as view categories.
* **User**
* Registration
* New User can create account by filling the following information such as first name, last name, contact, email, password, address.
* Login
* User can Login using his/her email/username and password.
* Search
  + The “Search” module enables users to find books on the website
* Category and Sub-Category
* This module allows user to browse books based on their categories and its sub-categories enhancing the user’s ability to find books of their interest.
* View Books
  + The “View Book” module provides detailed information about a specific book when selected by a user, including its title, author, price, stock, description.
* Cart
* The "Add to Cart" module allows users to add books they wish to purchase to a virtual shopping cart.
* It displays the selected books, their quantities, and the total price.
* Checkout
* It is the final step in purchasing process.
* It allows user to input shipping information, select a payment method and confirm their order.
* Contact
* User can submit their queries, share review of their books.
* User also share their email or username.
  1. **Features of proposed system**

These features can help create a robust and user-friendly ecommerce website :-

Admin Module:

1. Login: Admin can log in with their username and password.

2. Insert and View Book: Admin can add new books to the website and see the list of all available books.

3. Order List: Admin can view all orders made by users, including invoice number, date, total price, and status.

4. Category and Sub-Category: Admin can manage book categories and sub-categories, adding information as needed.

User Module:

1. Registration: New users can sign up by providing their name, contact info, email, password, and address.

2. Login: Registered users can log in with their email and password.

3. Search: Users can search for books on the website.

4. Category and Sub-Category: Users can browse books based on categories and sub-categories.

5. View Books: Users can see detailed information about specific books, including title, author, price.

6. Cart: Users can add books they want to purchase to a virtual shopping cart, which shows selected books, quantities, and total price.

7. Checkout: Users can finalize their purchases by entering selecting a payment method, and confirming their order.

8. Contact: Users can submit queries, share book reviews, and provide contact information like email or username.

**3. 6 Requirements Specification**

3.3.2 Hardware requirements:

Hardware requirements for running this project are as follows:

Processor: - Pentium I or above.

RAM: - 128 MB or above.

HD: - 20 GB or above.

3.3.3 Software requirements:

Software required to make working of product is:

Front end- HTML, PHP, CSS, JAVASCRIPT

Back end- MYSQL

Browser- Chrome or Microsoft Edge

Payment Gateway Integration: For processing transactions securely**.**

* 1. **Requirement Analysis**

A requirement specification document for an online book shopping e-commerce website should include both functional and non-functional requirements. Functional requirements describe what the system should do, while non-functional requirements describe how the system should perform.

Functional requirements that could be included in the document:

1. **User registration and login**: Users should be able to create an account and log in to the website.
2. **Search functionality**: Users should be able to search for books by title, author, or keyword.
3. **Product pages**: Each book should have its own product page with cover image, price.
4. **Shopping cart**: Users should be able to add books to their cart and view the contents of their cart.
5. **Checkout process**: Users should be able to enter their shipping and billing information and complete their purchase.

Non-functional requirements that could be included in the document:

1. **Performance**: The website should load quickly and be responsive to user input.
2. **Security**: User data should be stored securely and transactions should be encrypted.
3. **Scalability**: The website should be able to handle a large number of users and transactions.
4. **Compatibility**: The website should be compatible with a variety of devices and web browsers.
5. **Usability**: The website should be easy to navigate and use.
6. Fact-finding techniques are essential for gathering information effectively. Here's how each technique can be applied to gather information for an online book shopping ecommerce website:
   1. Observation:
   2. Observe user behavior on the website: Analyze how users navigate through the website, which pages they visit the most, and where they spend the most time.
   3. Monitor website performance: Look at metrics such as page load times, bounce rates, and conversion rates to identify any issues that may affect the user experience.
   4. Analyze competitor websites: Observe how other online bookstores design their websites, what features they offer, and how users interact with them.
   5. Record Review:
   6. Analyze website analytics: Review data from tools like Google Analytics to understand user demographics, traffic sources, popular books/categories, and conversion funnels.
   7. Study sales and inventory reports: Examine records of book sales, popular authors, bestselling genres, and inventory turnover rates to make informed decisions about stock management and marketing strategies.
   8. Review customer feedback and reviews: Look at customer reviews, ratings, and feedback to identify areas for improvement and understand customer preferences and pain points.
   9. Interview:
   10. Conduct interviews with website users: Talk to frequent users of the website to gather feedback on their experiences, preferences, and suggestions for improvement.
   11. Interview customer service representatives: Gather insights from customer service representatives about common customer inquiries, complaints, and issues encountered on the website.
   12. Interview stakeholders and employees: Speak with key stakeholders, such as the website owner, managers, developers, and marketers, to understand business goals, technical capabilities, and marketing strategies.
   13. Questionnaires:
   14. Design online surveys/questionnaires: Create surveys to gather quantitative data on user preferences, satisfaction levels, and feedback regarding website usability, book selection, pricing, and customer service.
   15. Distribute surveys to website visitors: Implement pop-up surveys or email surveys to collect feedback from users about their experiences with the website and their preferences for online book shopping.
   16. Analyze survey responses: Use survey data to identify trends, preferences, and areas for improvement, and to make data-driven decisions for optimizing the website and enhancing the user experience.

ii. Feasibility Study

1. Economical Feasibility:

* 1. Cost-Benefit Analysis: Determine the costs associated with developing and maintaining the website versus the potential benefits in terms of revenue generation.
  2. Revenue Projections: Estimate potential revenue based on market research, competitor analysis, and projected user traffic.
  3. Return on Investment (ROI): Calculate the expected ROI over a specific period, considering factors like development costs, operational expenses, and projected profits.
  4. Financial Viability: Assess the financial resources required to launch and sustain the website, including initial investment, operational costs, marketing expenses, and expected revenue streams.

2. Technical Feasibility:

* 1. Infrastructure Requirements: Evaluate the technical infrastructure needed to support the website, including web hosting, database management, and security measures.
  2. Scalability: Determine if the website can accommodate increasing user traffic and growing database size without compromising performance.
  3. Technology Stack: Select appropriate technologies and frameworks for website development, considering factors like scalability, security, and compatibility with desired features.
  4. Integration Capabilities: Assess the feasibility of integrating third-party services for payment processing, shipping, and other functionalities.
  5. Development Timeline: Estimate the time required for website development, testing, and deployment, considering factors like complexity of features and availability of resources.

3. Operational Feasibility:

* 1. User Acceptance: Evaluate the willingness of target users to adopt and use the website for purchasing books online, considering factors like convenience, user experience, and available alternatives.
  2. Resource Availability: Assess the availability of human resources, technical expertise, and operational support required to manage and maintain the website effectively.
  3. Regulatory Compliance: Ensure compliance with relevant regulations and standards related to e-commerce, data privacy, and consumer protection.
  4. Risk Management: Identify potential risks and challenges associated with website operations, such as cyber threats, data breaches, and system failures, and develop strategies to mitigate these risks.
  5. Continuous Improvement: Establish processes for gathering user feedback, monitoring website performance, and implementing updates and enhancements to improve user satisfaction and maintain competitiveness.

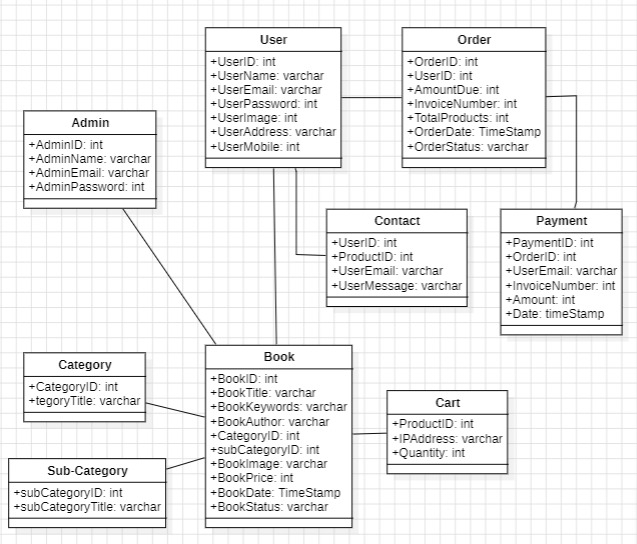
**Chapter 4**

**System Design**

* 1. **Basic Modules**
* **Admin**
* Login:
* The admin can log in using their credentials.
* Insert and View Book:
  + It displays a list of all the books available on the website, providing basic information such as the book title, author and also admin can add books.
  + It also displays entries .
* Order list
* It provides a comprehensive overview of all the orders placed by users.
* Each order record includes essential details such as invoice number, order date, total price and order status.
* Category and Sub-Category
  + The admin can add categories and its sub-categories as well as view categories.
* **User**
* Registration
* New User can create account by filling the following information such as first name, last name, contact, email, password, address.
* Login
* User can Login using his/her email/username and password.
* Search
  + The “Search” module enables users to find books on the website
* Category and Sub-Category
* This module allows user to browse books based on their categories and its sub-categories enhancing the user’s ability to find books of their interest.
* View Books
  + The “View Book” module provides detailed information about a specific book when selected by a user, including its title, author, price, stock, description.
* Cart
* The "Add to Cart" module allows users to add books they wish to purchase to a virtual shopping cart.
* It displays the selected books, their quantities, and the total price.
* Checkout
* It is the final step in purchasing process.
* It allows user to input shipping information, select a payment method and confirm their order.
* Contact
* User can submit their queries, share review of their books.
* User also share their email or username.
  1. **Schema Design**

• Database schema design refers to the strategies and practices for constructing a database schema.

• A database schema is a description of how data is structured or organized in a database.



**Notations**

 **Class :-** A class is a basic component in a UML class diagram. A class is represented by a rectangle. The name of the class comes on the top of the rectangle with a separator below. Below is a representation of a Person class-diagram

### **Class Attributes :-** The attributes are properties of class. Remember that class is a blue print These attributes correspond to member variables (data members) in the code.

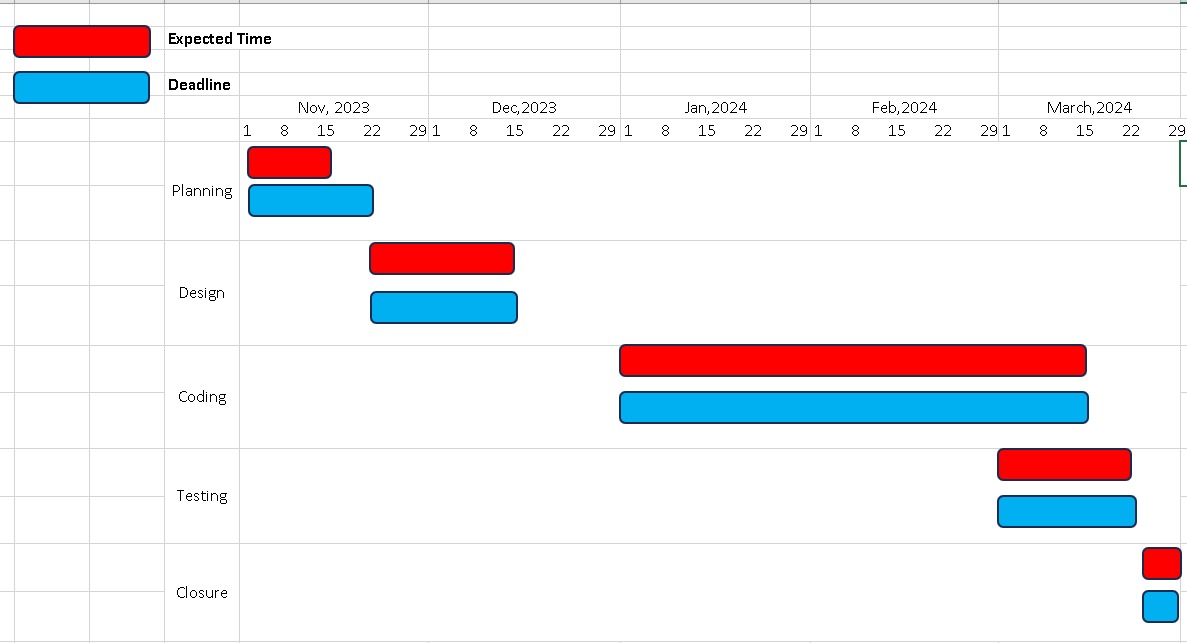
 **Associations :-** Associations are relationships between classes in a UML Class Diagram. They are represented by a solid line between classes.

* 1. **Gantt Chart**

A Gantt chart is a project management tool that illustrates work completed over a period of time in relation to the time planned for the work. It typically includes two sections: the left side outlines a list of tasks, while the right side has a timeline with schedule bars that visualize work.

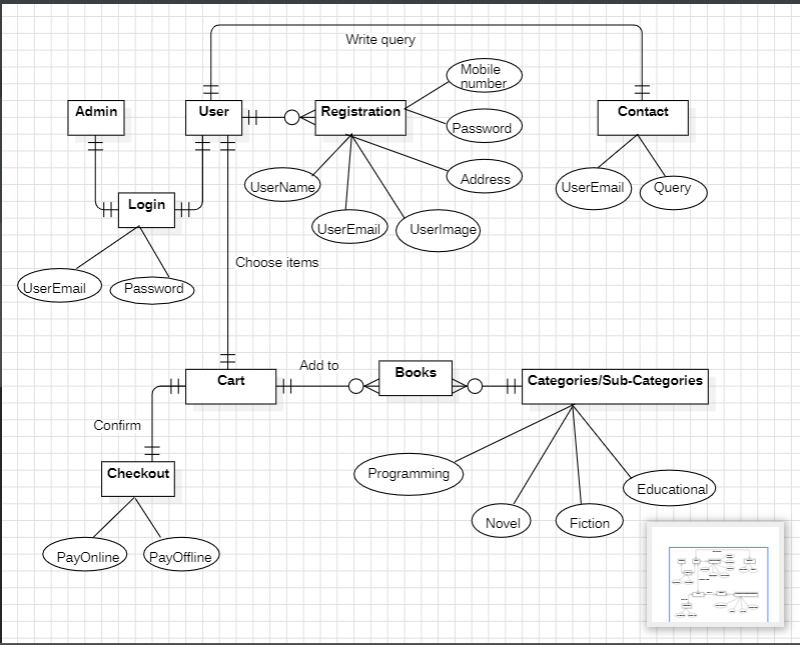
**Color Notation on gantt chart:**

|  |  |
| --- | --- |
| **Expected Time** |  |
| **Deadline** |  |

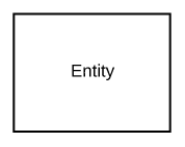


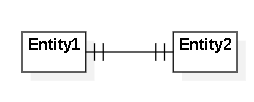
* 1. **ER-Diagram**

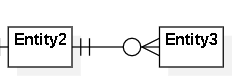
An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system.

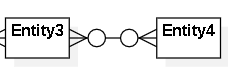


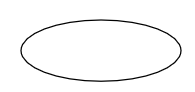
**Notations**

**Entity :-** These shapes are independent from other entities, and are often called parent entities, since they will often have weak entities that depend on them. They will also have a primary key, distinguishing each occurrence of the entity.

 **One to one relationship**

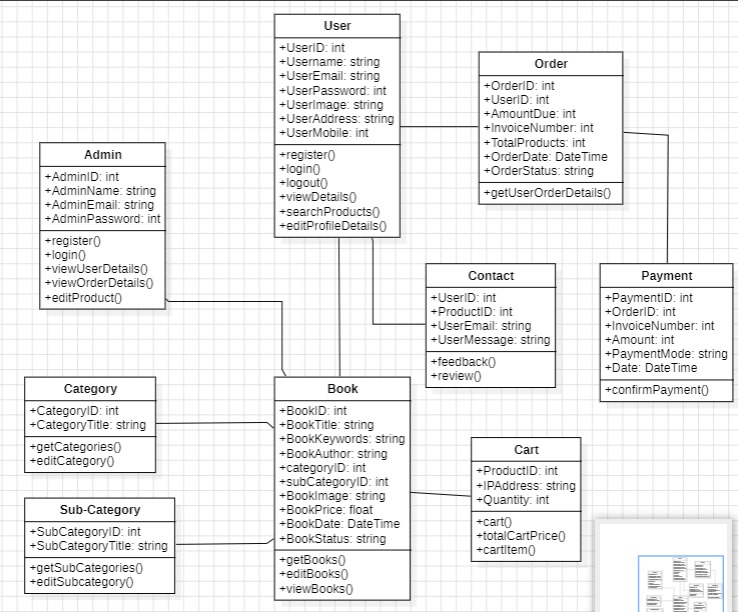
**One to many**

 **Many to many**

**Attribute :-** Attributes are characteristics of an entity, a many-to-many relationship, or a one-to-one relationship.

* 1. **Class Diagram**

Class diagrams are a type of UML (Unified Modeling Language) diagram used in engineering to visually represent the structure and relationships of classes within a system i.e. used to construct and visualize object-oriented systems.



**Notations**

 **Class :-** A class is a basic component in a UML class diagram. A class is represented by a rectangle. The name of the class comes on the top of the rectangle with a separator below. Below is a representation of a Person class-diagram

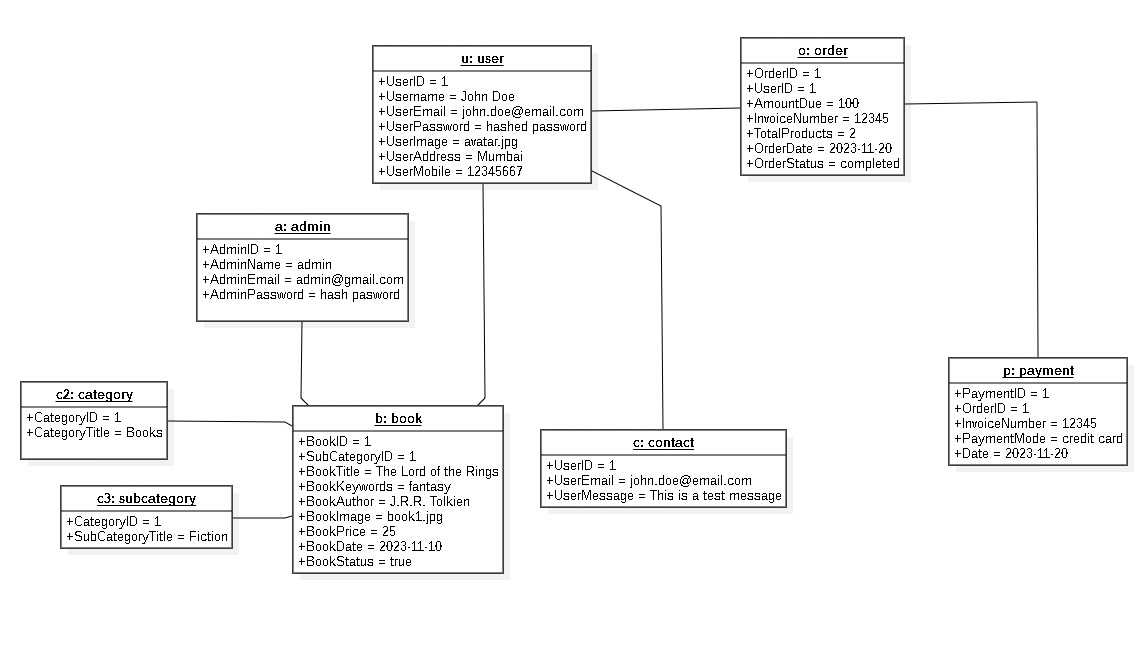
### **Class Attributes :-** The attributes are properties of class. Remember that class is a blue print These attributes correspond to member variables (data members) in the code.

**Class Operations :-** Operations are shown in the third partition. They are services the class provides.

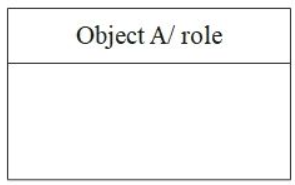
 **Associations :-** Associations are relationships between classes in a UML Class Diagram. They are represented by a solid line between classes.

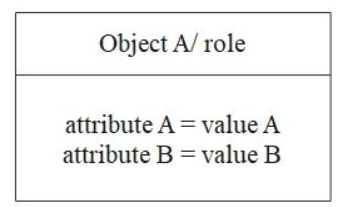
* 1. **Object Diagram**

Object diagrams are a visual representation in UML (Unified Modeling Language) that illustrates the instances of classes and their relationships within a system at a specific point in time. They display objects, their attributes, and the links between them, providing a snapshot of the system’s structure during execution.



**Notations**

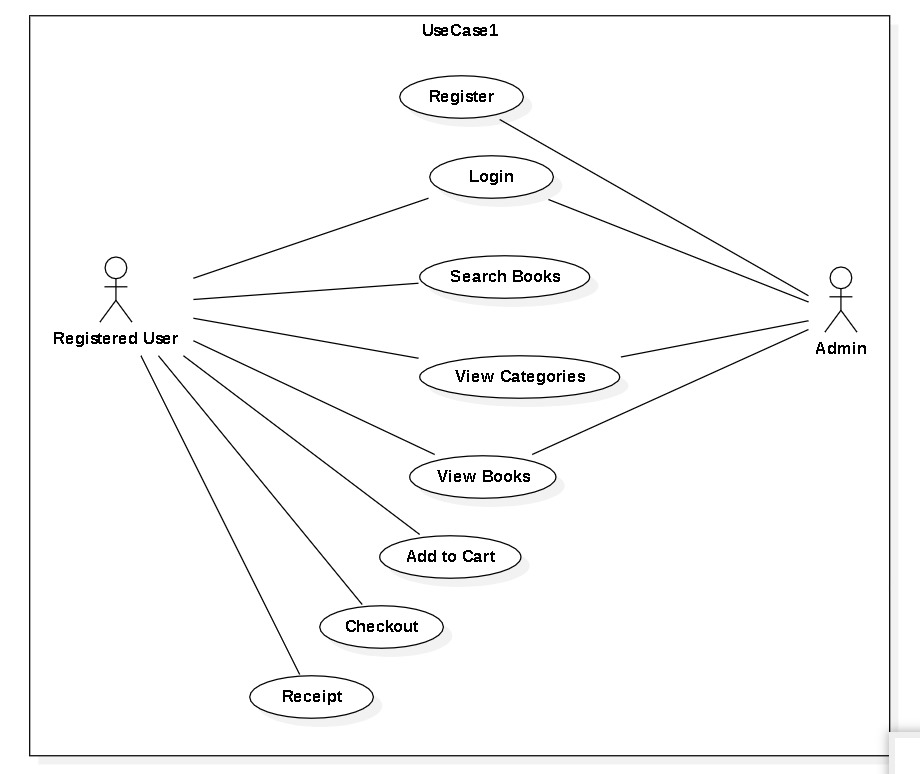
 **Object Name :-** An object is represented by a rectangular symbol, as shown, generally with two rows. The object name is shown at the top row of the rectangular symbol.

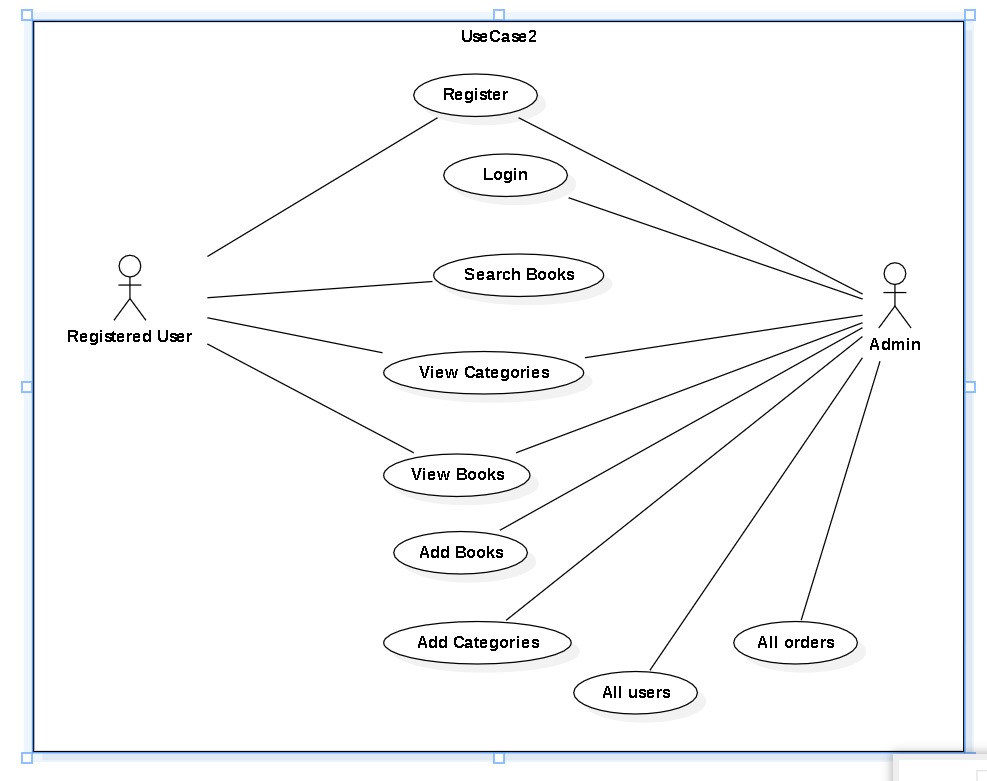
 **Attributes :-** Object attributes are shown in the bottom row of the rectangular symbol. They must have a value assigned to them.

 **Links :-** Use links when there are instances of association between objects.

* 1. **Use Case Diagram**

A Use Case Diagram is a type of Unified Modeling Language (UML) diagram that represents the interaction between actors (users or external systems) and a system under consideration to accomplish specific goals. It provides a high-level view of the system’s functionality by illustrating the various ways users can interact with it.





**Notations**

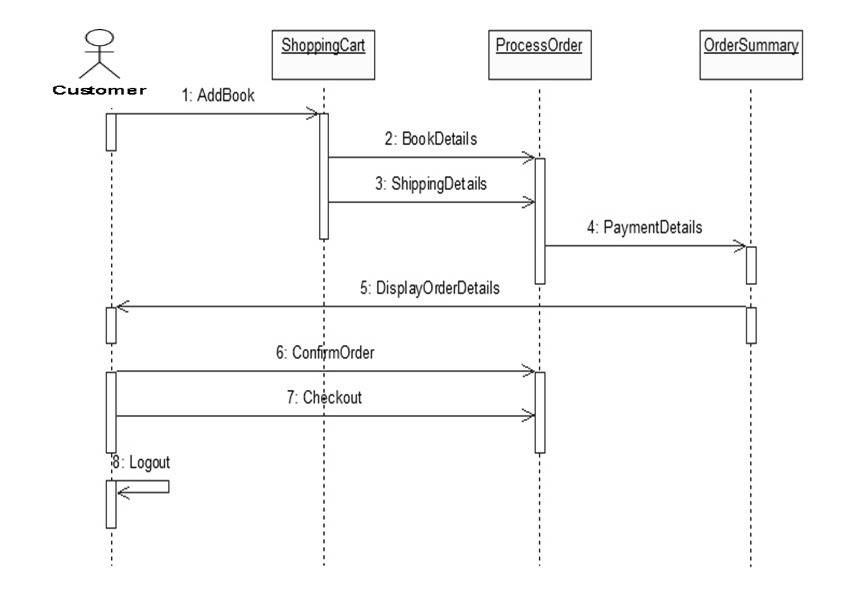
 **Actor :-** Stick figures that represent the people actually employing the use cases.

**use case** :- Horizontally shaped ovals that represent the different uses that a user might have.

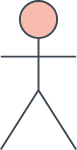
**Associations :-** A line between actors and use cases. In complex diagrams, it is important to know which actors are associated with which use cases.

* 1. **Sequence Diagram**

A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process.



**Notations**

** Actor :-** Shows entities that interact with or are external to the system.

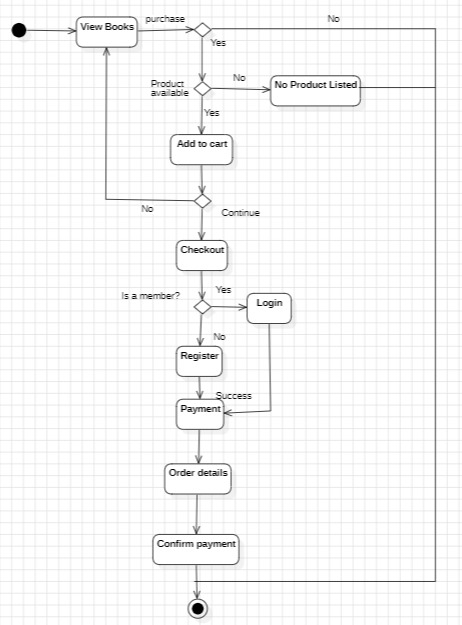
** Activation box** :- Represents the time needed for an object to complete a task. The longer the task will take, the longer the activation box becomes.

**Lifeline :-** Represents the passage of time as it extends downward. This dashed vertical line shows the sequential events that occur to an object during the charted process. Lifelines may begin with a labeled rectangle shape or an actor symbol.

 **Synchronous message** :- Represented by a solid line with a solid arrowhead. This symbol is used when a sender must wait for a response to a message before it continues. The diagram should show both the call and the reply.

* 1. **Activity Diagram**

An activity diagram is a type of Unified Modeling Language (UML) flowchart that shows the flow from one activity to another in a system or process. It's used to describe the different dynamic aspects of a system and is referred to as a 'behavior diagram' because it describes what should happen in the modeled system.

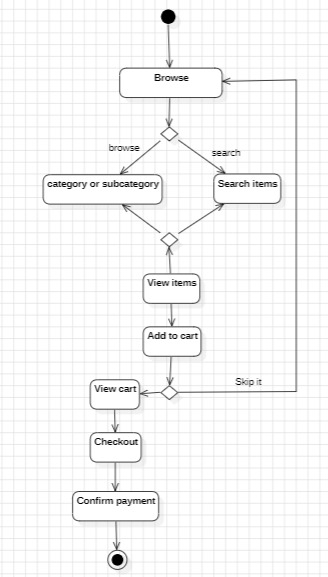


**Notations**

|  |  |  |  |
| --- | --- | --- | --- |
| start Symbol **Initial** :- 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 **Activity symbol** :- 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 **Control flow** :- 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**Decision symbol** :- 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 **Final** :- Marks the end state of an activity and represents the completion of all flows of a process. |  |  | |

* 1. **State Diagram**

A state diagram is used to represent the condition of the system or part of the system at finite instances of time. It’s a behavioral diagram and it represents the behavior using finite state transitions.



**Notations**

### State Diagram Symbols - First StateFirst state :- A marker for the first state in the process, shown by a dark circle with a transition arrow.

### State Diagram Symbols - State Shape State :- A rectangle with rounded corners that indicates the current nature of an object.

### State Diagram Symbols - Terminator Shape Final State :- A circle with a dot in it that indicates that a process is terminated.

### State Diagram Symbols - Choice PseudostateChoice :- A diamond symbol that indicates a dynamic condition with branched potential results.

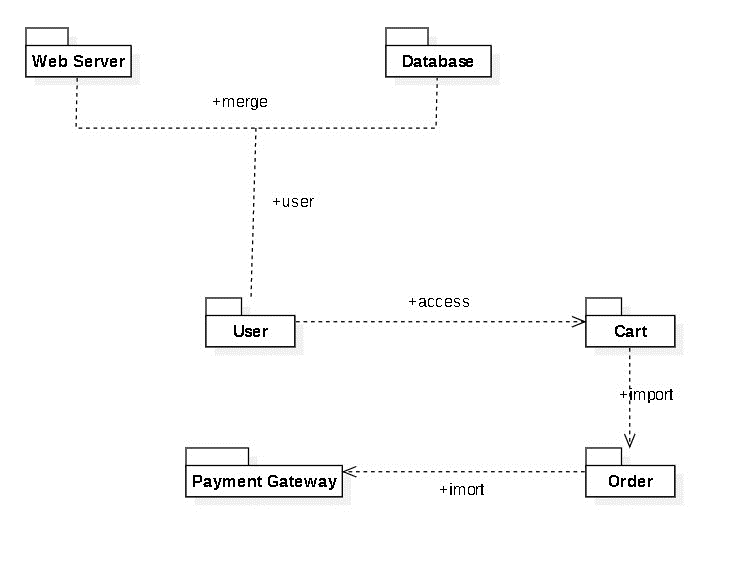
### State Diagram Symbols - Transition ShapeTransition :-

An arrow running from one state to another that indicates a changing state.

* 1. **Package Diagram**

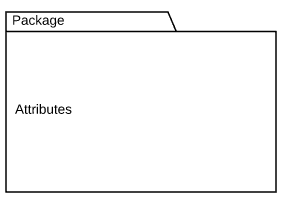
It groups related UML elements (classes, diagrams, etc.) into logical units called packages. Think of them as folders for your system's component

+import





**Notations**

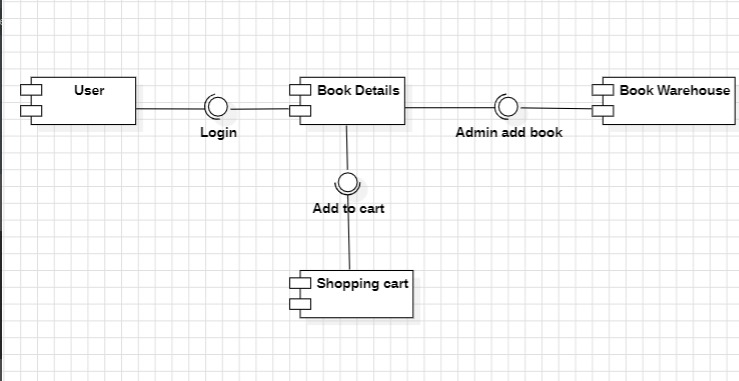
Package :- Groups common elements based on data, behavior, or user interaction

Dependency :- Depicts the relationship between one element (package, named element, etc) and another

Dependencies types

1. Merge: This is a relationship where a package can be merged with other packages1. It’s used to combine the contents of two packages into a new package1.
2. Use: This is a relationship that signifies that one thing uses another1. In the context of packages, it means that one package uses the functionalities of another package1.
3. **Import:** Indicates that functionality has been imported from one package to another.
4. **Access:** Indicates that one package requires assistance from the functions of another package.
   1. **Component Diagram**

A component diagram is a blueprint that shows how different parts of a system fit together. It's like a map of the system, highlighting the individual components and how they connect and depend on each other. These diagrams are especially useful in software development for visualizing complex systems.

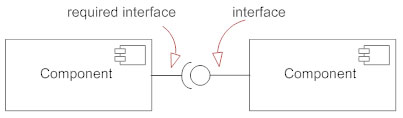


**Notations**



**Components:**

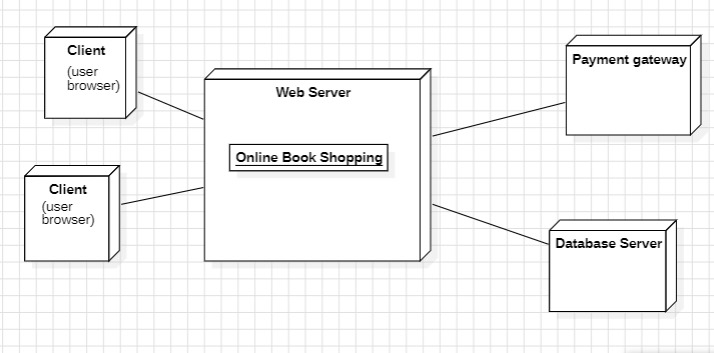
* Represented by rectangles.
* Each rectangle depicts a single, independent unit of the system with its functionalities encapsulated.



**Interfaces:**

* **Provided Interface (Lollipop symbol):** A circle connected to a line pointing outwards signifies an interface that a component offers functionalities for. Imagine it as a service the component provides to others.
* **Required Interface (Socket symbol):** A semicircle connected to a line pointing inwards signifies an interface that a component relies on to function. It represents functionalities the component needs from others.
  1. **Deployment Diagram**

A deployment diagram is a blueprint that shows how software components are physically deployed on hardware devices. It's like a map of your system's infrastructure, visualizing things like servers, databases, and the software that runs on them. These diagrams are part of the Unified Modeling Language (UML) and are useful for understanding complex systems.



**Notations**

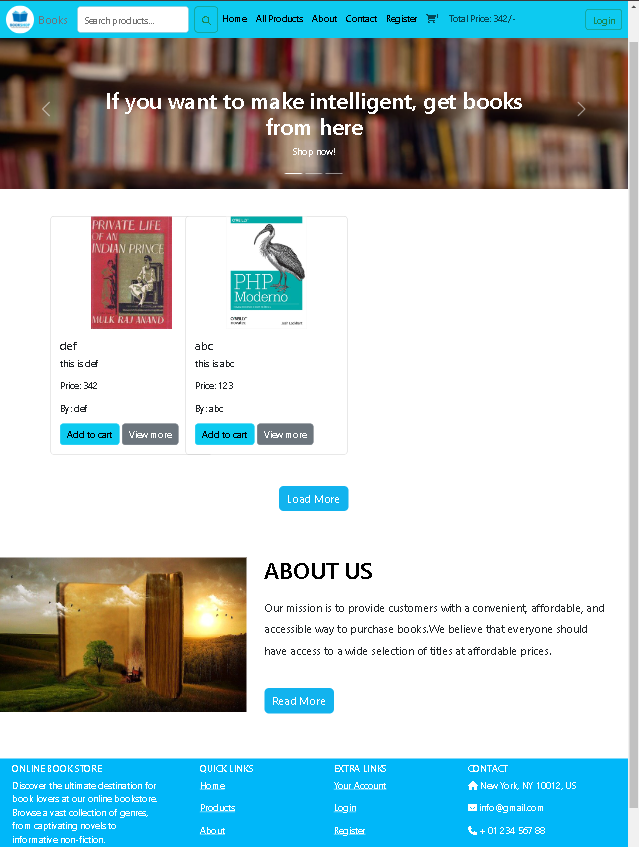
Deployment Diagram nodes

* **Node:** Represents a physical device where software components run. Think of it as a computer, server, sensor, or any device that executes parts of your system. It's typically shown as a 3D cube with a name.
* **Communication Path:** Shows how nodes exchange information. It's a line connecting two nodes and represents the network connection between them. You can optionally specify the type of protocol used (e.g., TCP/IP) on the path itself.

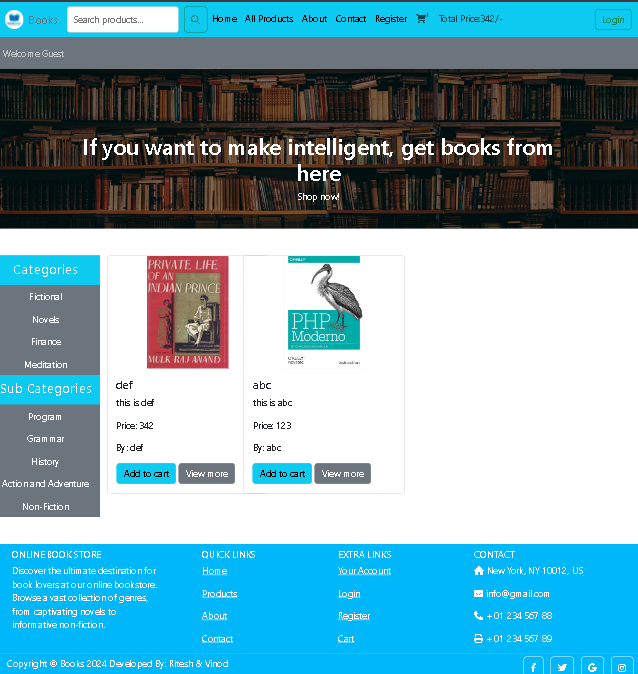
**Chapter 6**

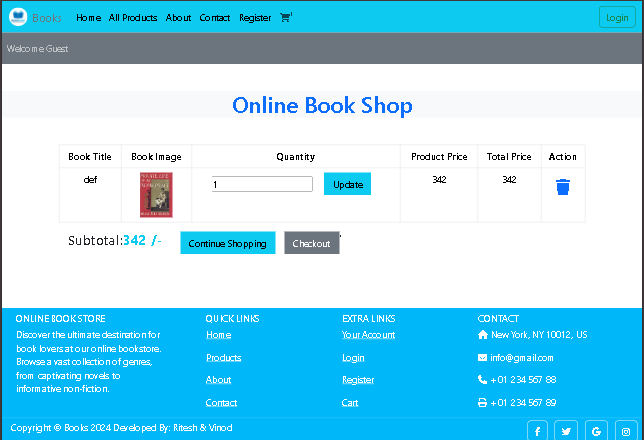
**Implementation and Testing**

**Index page**

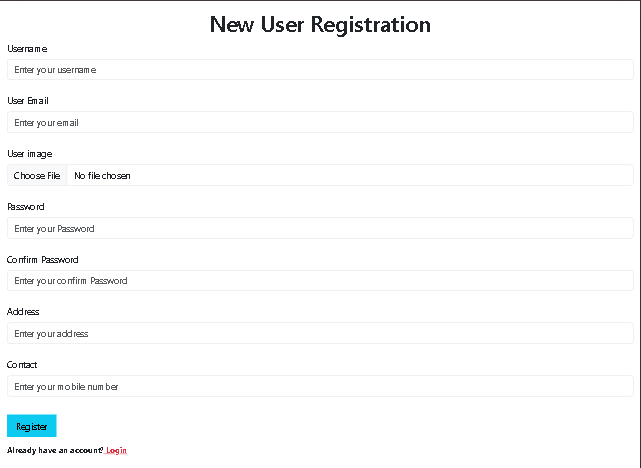
****

**Display all products page**

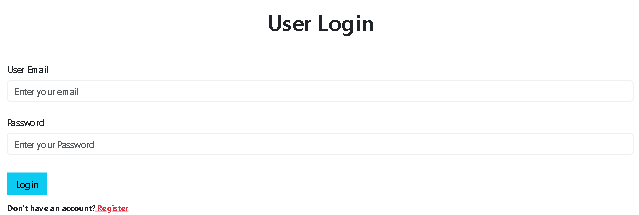


**Cart Page**

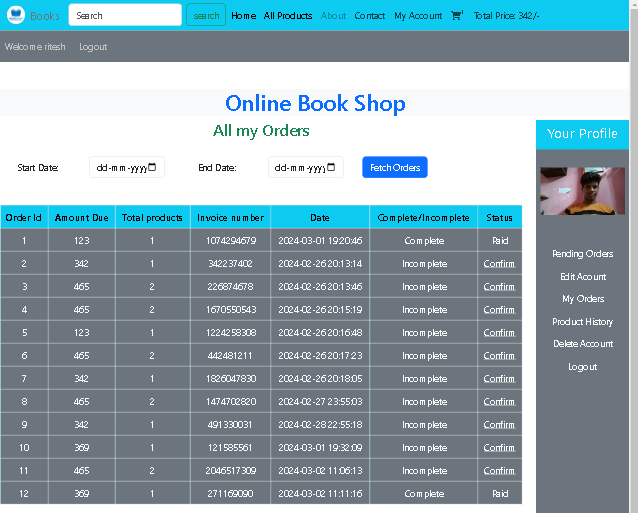
**Registration page**



**Login Page**



**Profile Page**



* 1. **Code (Place Core segments)**

**//index.php**

This page is a PHP and HTML mix for an online book shopping e-commerce website:

1. Initialization and Imports:

- PHP `include` statements for connecting to the database and common functions.

- HTML head section with meta tags, title, and links to Bootstrap CSS and Font Awesome.

2. CSS Styling:

- Custom styles for the logo, body, and responsive adjustments.

- Additional styling for a search button.

3. jQuery and AJAX Script:

- jQuery script for handling the autocomplete feature in the search bar.

- AJAX request to `autocomplete.php` on keyup to fetch and display search suggestions.

4. Navigation Bar:

- Bootstrap navbar with a logo, links, search bar, and user-related links.

- Dynamic display of “Register” or “My Profile” based on user session.

- Cart details and total price display.

5. User Session Handling:

- Checking if the user is logged in and displaying a personalized welcome message.

- Logout link for authenticated users.

6. Carousel Section:

- Bootstrap carousel displaying images with captions for promotional purposes.

7. Product Display:

- Function calls to retrieve and display products.

- “Load more” link to navigate to additional products.

8. About Us Section:

- Section with an image and content describing the mission of the website.

- “Read more” link to learn more about the mission.

9. Footer:

- Inclusion of the footer section with a separate file (`footer.php`).

10. Bootstrap JavaScript:

- Link to Bootstrap’s JavaScript for responsive features and carousel functionality.

**//display\_all.php**

This page includes the setup of the HTML structure, Bootstrap for styling, and some PHP logic for handling user sessions, search functionality, navigation, and displaying products.

1. Bootstrap Integration: You’ve integrated Bootstrap for styling, making your website responsive and visually appealing.

2. Search Functionality: AJAX is used for real-time search suggestions as the user types in the search bar. This enhances the user experience.

3. Navigation: The navigation bar includes links to various sections like Home, All Products, About, Contact, user authentication (Register/Login), and a shopping cart display.

4. User Session Handling: The code checks whether a user is logged in or not. If logged in, it displays a personalized welcome message with the username and a logout link.

5. Product Display: Products are displayed dynamically, and categories and subcategories are fetched from the database.

6. Footer Inclusion: The footer is included as a separate file, promoting code reusability.

**//Cart.php**

The code you provided is a snippet of the shopping cart functionality for an online book store website. Here's a breakdown of the key functionalities implemented in the code:

1. **Includes:**
   * The code starts by including necessary files like connect.php for database connection, common\_function.php for commonly used functions, and starts a PHP session using session\_start().
2. **Update Cart Quantities:**
   * It checks if the form is submitted with a name attribute of update\_cart.
   * If submitted, it loops through each product ID (update\_id) and its corresponding quantity (quantity) submitted from the form.
   * It validates the quantity to be a positive integer and updates the cart details table in the database with the new quantity using UPDATE query.
3. **Displaying Cart Items:**
   * The code retrieves cart details for the user based on their IP address using a SELECT query on the cart\_details table.
   * It checks if there are any items in the cart.
   * If there are items, it displays a table with columns for product title, image, quantity, product price, total price, and an action to remove the item.
   * To calculate the total price for each item, it multiplies the product price with the quantity.
   * A variable $total\_price is used to accumulate the total price of all items in the cart.
4. **Removing Items from Cart:**
   * The code checks if a GET parameter named remove is set in the URL.
   * If remove is set, it retrieves the product ID to be removed ($remove\_id).
   * It then deletes the product from the cart details table using a DELETE query based on the product ID.
   * Finally, it redirects the user back to the cart page using JavaScript's window.open function.
5. **Continue Shopping and Checkout Buttons:**
   * The code displays a "Continue Shopping" button and a "Checkout" button conditionally based on whether there are items in the cart.
   * Clicking the "Continue Shopping" button redirects the user to the homepage (index.php) using JavaScript if the form is submitted with the name continue\_shopping.
   * The "Checkout" button links to the checkout page (./user/checkout.php).
6. **Footer:**
   * The code includes the footer section of the website from a separate file footer.php.
7. **Bootstrap Integration:**
   * The code includes Bootstrap CSS and JavaScript libraries for responsive design and interactive elements.

.

**//user\_registration.php**

This page showcases the registration form and processing logic for a user registration system. Here's a breakdown of the core functionalities:

**HTML Form:**

* The code utilizes HTML to create a user registration form.
* It includes fields for username, email, profile image, password, confirm password, address, and contact number.
* Bootstrap is used for layout and styling purposes.

**Server-side Processing (PHP):**

* The code utilizes PHP to process form submissions and interact with the database.
* It checks if the "user\_register" button is submitted.
* It retrieves user information from the submitted form ($\_POST).
* It performs the following actions:
  + Hashes the password using password\_hash for secure storage.
  + Validates if username and email already exist in the database (user\_table).
  + Validates if passwords match (entered password and confirm password).
  + Uploads the profile image using move\_uploaded\_file if no validation errors occur.
  + Inserts the new user data into the user\_table if all validations pass.
* Finally, it checks if there are any existing cart items associated with the user's IP address (ip\_address).
  + If cart items exist, it logs the user in and redirects them to the checkout page.
  + Otherwise, it redirects the user to the homepage (index.php).

**//user\_login.php**

This page implements a user login functionality for an online book store website. Here's a breakdown of the key functionalities:

1. **HTML Structure:**
   * The code starts with basic HTML tags to define the document structure.
   * It includes Bootstrap CSS for responsive design and styling.
   * The central content area is a container divided into rows and columns.
   * Inside the column, a form is designed to capture user email and password for login.
2. **Login Form:**
   * The form uses email and password fields for user credentials.
   * The email field (user\_email) and password field (user\_password) have labels and placeholders for better user experience.
   * The form is submitted using a POST method.
   * A submit button with the value "Login" is included.
   * Below the form, there's a link to the user registration page (user\_registration.php).
3. **PHP Code:**
   * The PHP code starts by checking if the login button (user\_login) is submitted in the form.
   * It retrieves the user email ($user\_email) and password ($user\_password) from the submitted form data.
   * A database query ($select\_query) is prepared to select all records from the user\_table where the user\_email matches the provided email.
   * The query is executed using mysqli\_query($con, $select\_query), where $con is likely a connection variable established elsewhere (possibly in the included connect.php file).
4. **Verifying Login Credentials:**
   * The number of rows returned by the query ($row\_count) is used to check if a user with the entered email exists in the database.
   * If the user exists ($row\_count > 0), another query ($select\_query\_cart) is run to check if there are any items in the user's cart based on their IP address ($user\_ip).
   * If the credentials match (password\_verify($user\_password, $row\_data['user\_password'])), the user is logged in by setting the user\_email in the session ($\_SESSION['user\_email'] = $user\_email).
   * There are two scenarios for redirection after successful login:
     + If there's only one record for the user ($row\_count == 1) and the cart is empty ($row\_count\_cart == 0), the user is redirected to their profile page (profile.php).
     + Otherwise, the user is redirected to the payment page (payment.php), likely to process any existing items in their cart.
5. **Error Handling:**
   * If the user credentials are invalid (no matching email or incorrect password), an alert message is displayed using JavaScript (echo "<script>alert('Invalid Credentials')</script>";).

**//my\_profile.php**

The provided page appears to be a portion of the HTML, PHP, and CSS code for a user profile page within an online book shopping website. Here's a breakdown of the code highlighting the core functionalities:

Functionality Breakdown:

• User Information Display:

o The code retrieves user information like email and username from the session and displays a profile picture stored in the user\_images folder.

• Navigation bar:

o The code creates a dynamic navigation bar that changes based on whether a user is logged in or not. When logged in, it displays a welcome message with the username and options to edit account, view orders, manage products, and logout. For guests, it displays a login link.

• Order details:

o It includes a get\_user\_order\_details() function, likely defined elsewhere, to fetch and display the user's order details.

• Conditional Content:

o The code utilizes conditional statements (e.g., isset($\_GET['edit\_account'])) to include specific content sections based on user actions. For instance, if the user clicks on "Edit Account" in the navigation bar, it includes the edit\_account.php file responsible for editing account information. Similarly, it handles other actions like viewing orders, deleting accounts, and viewing product history.

Core Technologies Used:

• HTML: Provides the basic structure and content of the webpage.

• PHP:

o Manages user sessions to access user information.

o Connects to the database to retrieve user details and order information (likely using functions defined elsewhere).

o Controls what content to display based on user actions using conditional statements.

• CSS: Styles the layout and visual appearance of the webpage elements.

* 1. **Testing approaches (Black and White box testing) :-**

Black Box Testing and White Box Testing are two fundamental approaches to software testing, each with its own objectives, techniques, and applications.

1. Black Box Testing:

- Objective: Black Box Testing focuses on validating the functionality of a software system without considering its internal structure or implementation details.

- Techniques: Testers design test cases based on the specifications, requirements, and expected behavior of the software.

- Advantages:

- Testers do not need to have knowledge of the internal code, making it suitable for testing by individuals who are not developers.

- It ensures that the software meets the specified requirements and behaves as expected from the end user's perspective.

- Disadvantages:

- It may not cover all possible scenarios since testers rely solely on specifications and requirements.

- It might be challenging to identify the root cause of defects, as testers lack insight into the internal workings of the software.

2. White Box Testing:

- Objective: White Box Testing focuses on verifying the internal logic, structure, and code of the software.

- Techniques: Testers design test cases based on the internal workings of the software, including code paths, branches, and conditions.

- Advantages:

- It provides insights into the code's structure, helping to identify potential vulnerabilities and areas of improvement.

- Test coverage can be more comprehensive as testers can target specific code paths and conditions.

- Disadvantages:

- Testers conducting white box testing typically require programming knowledge and access to the source code, making it less accessible to non-developers.

- It may focus more on how the code works rather than whether it meets the end user's requirements.

In practice, both Black Box and White Box Testing are essential for ensuring software quality. They complement each other, with Black Box Testing validating the software against requirements and user expectations, while White Box Testing verifies the internal mechanisms and code quality. A combination of these testing approaches, known as Grey Box Testing, can also be employed to leverage the benefits of both techniques.

* 1. **Unit Testing: Boundary Value Analysis**

Boundary Value Analysis (BVA) is a software testing technique used to identify test cases that focus on the boundaries of input ranges, rather than the arbitrary values within those ranges. It is particularly effective in finding defects related to boundary conditions. Unit testing with Boundary Value Analysis involves testing the boundaries of input values for a particular unit of code to ensure that the behavior is correct.

Let's apply Boundary Value Analysis (BVA) to an online book shopping scenario for testing the "Updating quantity" functionality:

**Project:** Online Book Shopping - Add to Cart includes updating quantity functionality

**Test Case**

* Validate system behaviour when adding a book to the cart with various quantities.

**Input Values and Boundaries:**

* **Quantity:** Valid range is 1 to 100 (assumed limit based on business logic).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Cases:** Test Case ID | Input Quantity | Expected Result | Boundary Value | Result |
| TC-01 | 0 | Error: Invalid quantity | Minimum value - 1 below minimum | Fail |
| TC-02 | 1 | Book added to cart | Valid value within boundary | Pass |
| TC-03 | 100 | Book added to cart | Maximum value | Pass |
| TC-04 | 101 | Error: Invalid quantity | Maximum value + 1 | Fail |
| TC-05 | Non-numeric characters (e.g., abc) | Error: Invalid quantity | Invalid input type | Fail |

**Explanation:**

* TC-01 and TC-04 test adding zero or exceeding the maximum quantity, which are invalid scenarios.
* TC-02 and TC-03 test adding the minimum, maximum, and a valid quantity within the allowed range.
* TC-05 tests adding non-numeric characters, which is an unexpected input.

**Expected Results:**

* The system should display an error message for invalid quantities (TC-01, TC-04, TC-05).
* The system should successfully add the book to the cart for valid quantities (TC-02, TC-03).
  1. **Equivalence class Partitioning : -**

Equivalence Class Partitioning (ECP), also known as Equivalence Partitioning, is a software testing technique used in black-box testing. It helps design efficient test cases by dividing the input data into logical partitions called equivalence classes.

We can apply Equivalence Class Partitioning (ECP) to the user\_email field in your login form to identify potential test cases.

|  |  |  |  |
| --- | --- | --- | --- |
| Equivalence Class | Description | Test Case | Expected Result |
| Valid Email | Email address with a valid format (e.g., [email address removed]) and less than 100 characters. | [email address removed] | Login successful (if password is correct). |
| Empty Email | No email address entered in the field. |  | Alert: "Please enter your email". |
| Invalid Characters | Email address containing characters not allowed in email format (e.g., @#$%). | user@@domain | Alert: "Invalid email format". |
| Email exceeding length | Email address exceeding the 100 character limit. | [email address removed] | Alert: "Email address too long". |

**Explanation:**

* **Valid Email:** This class represents valid email addresses that adhere to the standard format and stay within the 100 character limit. The test case uses a simple email format to verify the system accepts correct emails.
* **Empty Email:** This class covers the scenario where the user leaves the email field blank. An empty email should trigger an alert prompting the user to enter their email.
* **Invalid Characters:** This class includes email addresses containing characters not allowed in a standard email format. The test case uses an example with double "@" symbols to ensure the system rejects such invalid formats.
* **Email exceeding length:** This class represents emails exceeding the 100 character limit. The test case uses a long string exceeding 100 characters to verify the system handles emails that violate the length restriction.