COMPREHENSIVE E-COMMERCE PLATFORM:

A FULL STACK PROJECT

THE DESIGN AND IMPLEMENTATION OF AN

E-COMMERCE SITE FOR ONLINE BOOK SALES

BTECH IV Semester

Computer Science and Engineering

BY

Yogeswari Borredi

Project Comprehensive E-Commerce Platform A Full Stack Project Submitted to the faculty of the Mjr college of Engineering and Technology

In partial fulfillment of the requirements for the degree computer science in the Department of Computer Science and Engineering.



**CERTIFICATE**

Certified that this is a bonafide record of the project work titled

COMPREHENSIVE E-COMMERCE PLATFORM: A FULL STACK PROJECT Done By:

B. Yogeswari

N. Mohan

B. Sudharshan

P. Dhanush

B. Surendra

G. Mounika

S. Prabhu Kumar

of IVsemester BTECH in the year 2024 in

partial fulfillment of the requirements for the award of Degree of

Bachelor of Computer Science & Engineering

Mrs. SOUNDARYA Mr. SURESH **Project Guide**  **Head of the Division**

**TABLE OF CONTENTS**

**INTRODUCTION------------------------------------------------------- 6**

**ABSTRACT----------------------------------------------------------- 7-8**

**ACKNOWLEDGEMENT-------------------------------------------- 9**

**PROJECT REPORT----------------------------------------------------- 10-21**

**CHAPTER-1: INTRODUCTION**

**LITERATURE REVIEW---------------------------------------------------21-24**

* 1. **– PROJECT AIMS & OBJECTIVES------------------------25**
  2. **– BACKGROUND OF PROJECT----------------------------26-27**

**CHAPTER-2: SYSTEM ANALYSIS---------------------------------------28**

**2.1 – SOFTWARE REQUIREMENT SPECIFICATION**

**2.1.1—PRODUCT DESCRIPTION------------------------------28-30**

**2.1.2 – SYSTEM OBJECTIVES-----------------------------------31**

**2.1.3—SYSTEM REQUIREMENTS-----------------------------32-34**

**2.1.4—SOFTWARE AND HARDWARE REQUIREMENTS—34-38**

**2.3—SOFTWARE TOOLS USED---------------------------------42**

**CHAPTER-3: PROJECT DESIGN---------------------------------------- 43**

**3.1—DATABASE DESIGN-----------------------------------------43-47**

**3.2—PROCESS MODEL-------------------------------------------47**

**3.2.1—FUNCTIONAL DECOMPOSITION DIAGRAM-------48**

**3.2.2—DATA FLOW DIAGRAMS--------------------------------49-56**

**3.3—INTERFACE DESIGN ----------------------------------------57-59**

**4—IMPLEMENTATION TECHNOLOGIES-----------------------60**

**4.1—INTERNET INFORMATION SERVICES--------------------61**

**4.2—ASP.NET---------------------------------------------------------62**

**4.2.1—AUTHENTICATION IN ASP--------------------------------64**

**4.3—MY SQL DATA BASE-------------------------------------------65**

**4.4—INTEGRATING IIS AND ASP.NET----------------------------66**

**4.5—INTEGRATING THE WEBSITE AND DATABASE-----------67**

**5—WEB PAGE PROGRAMMING OPTIONS-----------------------68**

**5.1—SERVERSIDE PROCESSING--------------------------------69-70**

**5.2—CLIENTSIDE PROCESSING------------------------------------71**

**6—WEB BASED APPLICATION DEVELOPMENT-----------------72**

**7—DATABASE CONNECTIVITY**

**7.1—ADO.NET-------------------------------------------------------73-76**

**7.2—CONNECTING ASP.NET--------------------------------------78**

**8—THE SHOPPING CART APPLICATION-----------------------79**

**8.1—SEARCH FOR BOOKS----------------------------------------80**

**8.2—REGISTRATION -----------------------------------------------81**

**8.3—USER DETAILES------------------------------------------------82**

**8.4—SHOPPING CART----------------------------------------------83**

**8.5—PLACE AN ORDER--------------------------------------------84-85**

**8.6—CHECK OUT----------------------------------------------------86-88**

**8.7—PURCHASE HISTORY----------------------------------------89-92**

**8.8—TRANSACTIONS----------------------------------------------93-96**

**9—LIMITATIONS AND FUTURE DEVELOPMENT---------------97 10—CONCLUSION-----------------------------------------------------98 11—BIBLIOGRAPHY------------------------------------------------99-100 12—WEBSITES--------------------------------------------------------101**

**FIGURE 1 ENTITY RELATIONSHIP DIAGRAM(ERD)………………………42**

**FIGURE 2 FUNCTIONAL DECOMPOSITION DIAGRAM…………………49**

**FIGURE 3 CUSTOMER - BROWSE CONTEXT DFD……………………….51**

**FIGURE 4 CUSTOMER - BROWSE DETAILED DFD...........................52**

**FIGURE 5 CUSTOMER – SHOPPING CART CONTEXT DFD................52**

**FIGURE 6 CUSTOMER - SHOPPING CART DETAILED DFD................53**

**FIGURE 7 CUSTOMER - SHOPPING CART DETAILED DFD................53**

**FIGURE 8 CUSTOMER – AUTHENTICATION – USERPROFILE DFD…54**

**FIGURE 9 AUTHENTICATED USER-PURCHASE CONTEXT DFD.........54**

**FIGURE 10 CUSTOMER - AUTHENTICATION - PURCHASE DFD.......55**

**FIGURE 11 CUSTOMER - NEWUSERREGISTRATION DFD.................56**

**FIGURE 12 ADMINSTRATOR CONTEXT DFD..................................57**

**FIGURE 13 ADMINISTRATOR DETAILED DFD..................................57**

**FIGURE 14 DISPLAY OF BOOKS PRESENT IN THE STORE................57**

**FIGURE 15 FOR SEARCHING THE BOOKS IN THE STORE...............58**

**FIGURE 16 SHOPPING CART FOR THE USER..................................59**

**FIGURE 17 REGISTRATION OF THE NEW USER..............................60**

**FIGURE 18 AUTHENTICATION OF THE USER.................................60**

**FIGURE 19 RELATION BETWEEN IIS AND ASP.NET.........................61**

**FIGURE 20 WEB PAGE PROGRAMMING OPTIONS........................68**

**FIGURE 21 COMPILED SERVER PROGRAMS FLOWCHART..............69**

**FIGURE 22 ADO.NET ARCHITECTURE.........................................76**

**FIGURE 23 BOOK DETAILS...........................................................79**

**FIGURE 24 SERACH FOR BOOKS.................................................80**

**FIGURE 25 NEW USER REGISTRATION........................................ 81**

**FIGURE 26 USER DETAILS.............................................................82**

**FIGURE 27 SHOPPING CART.........................................................83**

**FIGURE 28 ORDER DETAILS.........................................................84**

**FIGURE 29 SHIPPING DETAILS......................................................85**

**FIGURE 30 CHECK OUT................................................................86**

**FIGURE 31 ORDER CONFIRMATION............................................87**

**FIGURE 32 UPDATED INVENTORY AFTER ORDER PLACEMENT......87**

**FIGURE 33 PURCHASE HISTORY....................................................88**

**FIGURE 34 BOOK DETAILS.............................................................89**

**FIGURE 35 ADMINISTRATOR - MODIFY BOOKS............................90**

**FIGURE 36 DETAILS ABOUT NEW BOOK........................................91**

**FIGURE 37 UPDATED INVENTORY..................................................92**

**Design and Implementation of E-Commerce Site for Online shopping**

**INTRODUCTION** The business-to-consumer aspect of electronic commerce (e-commerce) is the most visible business use of the World Wide Web. The primary goal of an e-commerce site is to sell goods and services online. This project deals with developing an e-commerce website for Online Book Sale. It provides the user with a catalog of different books available for purchase in the store. In order to facilitate online purchase a shopping cart is provided to the user. The system is implemented using a 3-tier approach, with a backend database, a middle tier of Microsoft Internet Information Services (IIS) and ASP.NET, and a web browser as the frontend client. In order to develop an e-commerce website, a number of Technologies must be studied and understood. These include multi-tiered architecture, server and clientside scripting techniques, implementation technologies such as ASP.NET, programming language (such as C#, VB.NET), relational databases (such as MySQL, Access). This is a project with the objective to develop a basic website where a consumer is provided with a shopping cart application and also to know about the technologies used to develop such an application. This document will discuss each of the underlying technologies to create and implement an e-commerce website.

**ABSTRACT**

E-commerce has revolutionized the way people shop, offering convenience, accessibility, and a wide array of products at their fingertips. "E-Commerce360" aims to create a comprehensive e-commerce platform to cater to the evolving needs of online shoppers and businesses alike. This full stack project integrates various functionalities to provide a seamless shopping experience while empowering merchants with robust management tools.

**KEY FEATURES:**

**User Management:**

Registration and login for customers, enabling personalized experiences.

Profile management to update personal information, addresses, and payment methods.

**Product Catalog:**

Extensive product listings categorized for easy navigation.

Advanced search and filtering options for precise product discovery.

**Shopping Cart and Checkout:**

Secure shopping cart to add and manage selected items.

Smooth checkout process with multiple payment gateways for flexibility.

**Order Management:**

Order history and tracking to monitor the status of past and current orders.

Integration with shipping providers for real-time shipping updates.

**Merchant Dashboard:**

Registration and onboarding for sellers with verification processes.

Product management tools to add, edit, and remove listings efficiently.

**Analytics and Reporting:**

Sales analytics dashboard for both customers and merchants.

Detailed reports on sales, inventory, and customer behavior for informed decision-making.

**Customer Support:**

Help desk functionality for customer queries and issue resolution.

Integration with live chat and ticketing systems for real-time assistance.

**Technology Stack:**

**Frontend:** HTML, CSS, JavaScript, React.js

**Backend:** Node.js, Express.js, MongoDB

**Authentication**: JSON Web Tokens (JWT)

**Customer Support:** LiveChat API, Zendesk Integration

**Payment** Integration: Stripe, PayPal

**Analytics**: Google Analytics, Custom Dashboard Development

**ACKNOWLEDGEMEMNT**

In completing this graduate project I have been fortunate to have help, support and encouragement from many people. I would like to acknowledge them for their cooperation.

First, I would like to thank Mrs. Soundarya mam, my project advisor, for guiding me through each and every step of the process with knowledge and support. Thank you for your advice, guidance and assistance.

I would also like to thank my project committee members, who showed immense patience and understanding throughout the project and provided suggestions. help throughout the project.

Finally, I would like to dedicate this project to my parents, my husband kiran kumar and my brother sudharshan and my friends Mohan and Mounika, for their love, Encouragement.

**PROJECT REPORT**

**E-Commerce Industry**

E-commerce (electronic commerce) is the activity of electronically buying or selling of products on online services or over the Internet. The term was coined and first employed by Dr. Robert Jacobson, Principal Consultant to the California State Assembly's Utilities & Commerce Committee, in the title and text of California's Electronic Commerce Act, carried by the late Committee Chairwoman Gwen Moore (D-L.A.) and enacted in 1984. Electronic commerce draws on technologies such as mobile commerce, electronic funds transfer, supply chain management, Internet marketing, online transaction processing, electronic data interchange (EDI), inventory management systems, and automated data collection systems.

E-commerce is in turn driven by the technological advances of the semiconductor industry, and is the largest sector of the electronics industry. Modern electronic commerce typically uses the World Wide Web for at least one part of the transaction's life cycle although it may also use other technologies such as e-mail. Typical e-commerce transactions include the purchase of online books (such as Amazon) and music purchases (music download in the form of digital distribution such as iTunes Store), and to a less extent, customized/personalized online liquor store inventory services. There are three areas of e-commerce: online retailing, electronic markets, and online auctions. E-commerce is supported by electronic business. E-commerce businesses may also employ some or all of the followings:

• Online shopping for retail sales direct to consumers via Web sites and mobile apps, and conversational commerce via live chat, chatbots, and voice assistants;

• Providing or participating in online marketplaces, which process thirdparty business-to-consumer (B2C) or consumer-to-consumer (C2C) sales;

• Business-to-business (B2B) buying and selling;

• Gathering and using demographic data through web contacts and

Social media;

• Business-to-business (B2B) electronic data interchange; 2

• Marketing to prospective and established customers by e-mail or

(for example; with news letters)

• Engaging in pretail for launching new products and services;

• Online financial exchanges for currency exchanges or trading purposes.

**Advantages of a full-stack E-Commerce platform:**

1. **Customization**: Tailor-made to match business needs and branding.
2. **Scalability**: Easily grows with business, handling increased traffic and transactions.
3. **Integration**: Seamless integration with third-party services enhances functionality.
4. **Performance**: Optimization ensures fast loading times and smooth user interactions.
5. **Security**: Robust measures at every layer protect against threats and ensure safe transactions.
6. **Flexibility**: Ability to choose the best technologies for each layer of the application stack.
7. **Control**: Full control over development process, timeline, and feature prioritization.
8. **Cost-effectiveness**: Long-term savings compared to pre-built solutions with recurring fees.
9. **Ownership**: Complete control of codebase, data, and intellectual property.

**10.Competitive advantage**: Offers differentiation, unique features, and adaptability in the market.

Top of Form

**The potential disadvantages of a full-stack E-Commerce Platform:**

**1.Complexity**: Multi-layered development increases multiple

Complexity.

**2.Time-consuming**: Custom development takes longer than pre-built solutions.

**3.Higher upfront costs**: Initial investment in resources can be significant.

**4.Maintenance overhead**: Requires ongoing effort to maintain and update.

**5.Security risks**: Managing security across layers is critical.

**6.Dependency on expertise**: Reliance on internal developers for maintenance.

**7.Feature gaps**: May lack some features compared to pre-built solutions.

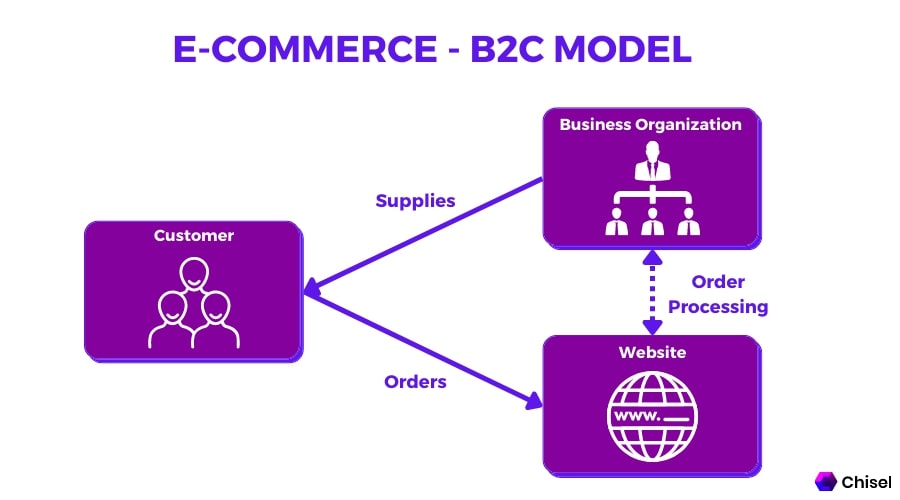
**8.Limited support**: Internal resources handle troubleshooting.

**9.Longer time to market**: Development may take more time to launch.

**10. Risk of over-engineering**: Potential for unnecessary complexity and costs.

Traditional Ecommerce business models typically fall into four main categories:

**Business-to-Consumer (B2C)**: This is the most common model, where businesses sell products or services directly to individual consumers. Examples include online retail stores like Amazon, clotthing brands with online shops. And digital content providers.



**Business-to-business(B2B):**

Business-to-business, or B2B, refers to commerce between two businesses rather than between a business and an individual consumer. It is one involving a manufacturer and wholesaler, or a wholesaler and a retailer.



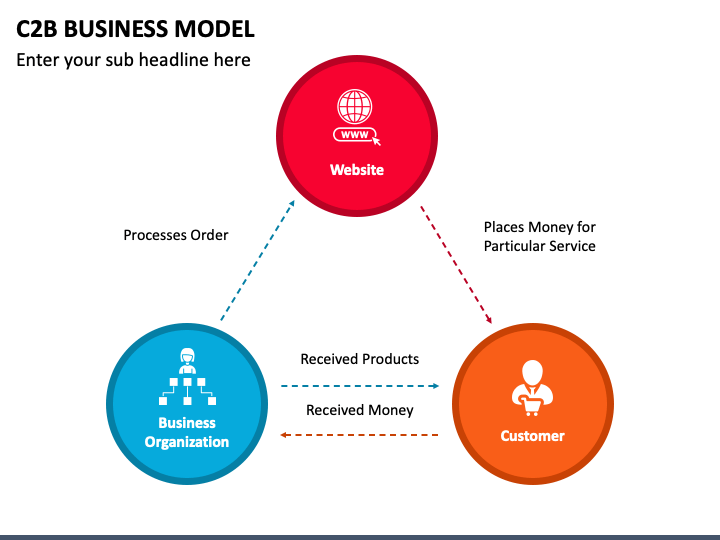
**Consumer to consumer(C2C):**

C2C also called as consumer to consumer commerce, is a business model often defined as one consumer selling goods or services to another consumer. When done online, C2C becomes a type of eCommerce.

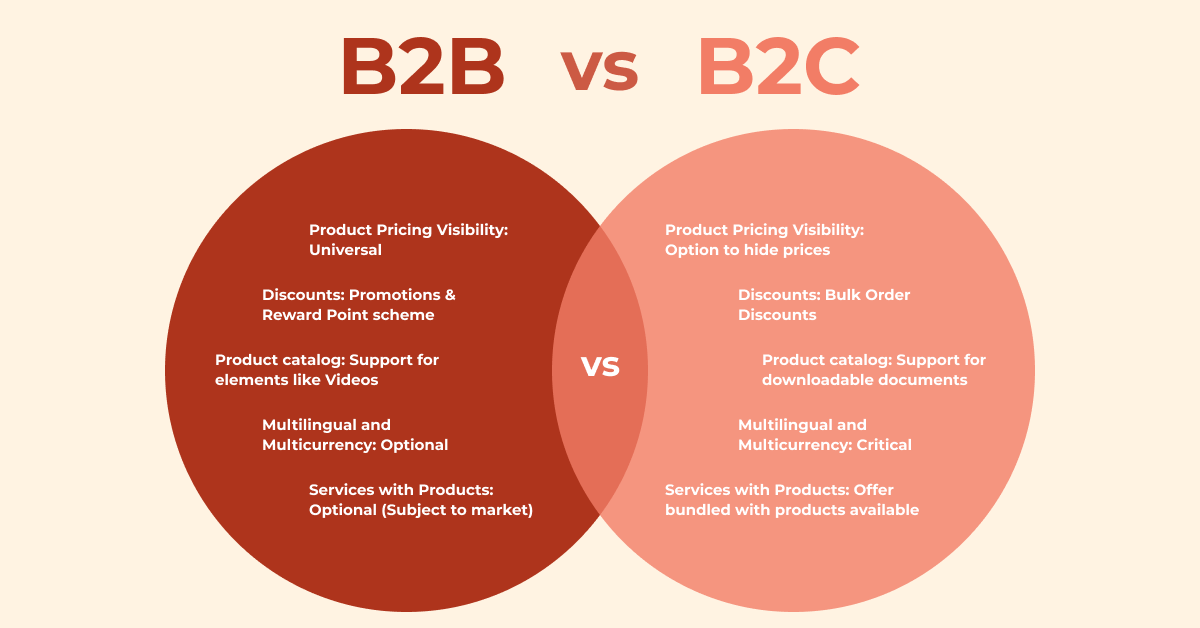


**Consumer to business(C2B):**

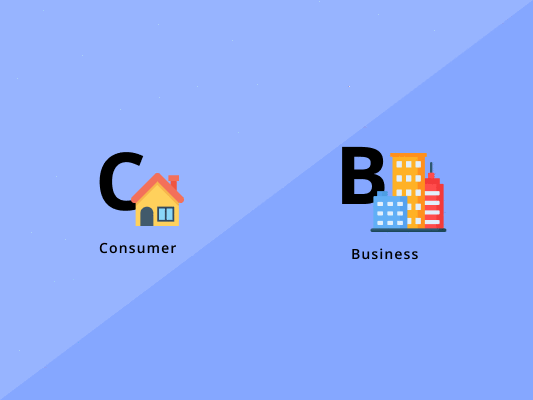
**C2B** is also refferd as consumer to business is an up and coming business market that can be utilized as a company's entire business model or added to an already existing model.



**DIFFRENCE BETWEEN B2B AND B2C**



**DIFFRENCE BETWEEN C2B AND C2C**



**OBJECTIVE OF ANALYSIS**

The major objectives of the project work on E-commerce industry are:

1.The E-Commerce goals focus on optimizing online retail performance. These goals typically include increasing website traffic, enhancing customer engagement, boosting conversion rates, and driving sales growth.

2. To understand about E-commerce and its role in the economy

3. To understand the types of E-commerce and its functioning

4. To understand the influence of the coronavirus pandemic on the E-Commerce Industry with respect to the consumers perspective.

5. To understand major academic disciplines contributing to e-commerce research

6.To understand about the growth prospects of E-commerce



**CHALLENGES IN ANALYSIS**

There were few challenges that were been faced during the project work on the E commerce Industry: -

• One of the major challenges during the analysis was the collection of primary data (original data) related to the study.

• Managing vast and varied data streams.

• Searching the accurate data related to the study was another challenge.

• Knowing the appropriate perspective and the preparation of questionnaire was also one of the challenges.

• Studying the impact of covid on e-commerce was a challenge in terms getting the data.

• Integration data from multiple channels.

• Providing real-time insights for personalized experiences.

• Understanding complex user behaviour patterns.

• Ensuring data accuracy and consistancy.

**Methodology:**

This paper used a survey of 67 people on their perspective as a consumer involved in online shopping. 9 Questions were asked to them for a clearer picture of the viewpoint changes of consumer during the pandemic of covid-19 towards E-Commerce. The Thought process of the world is now changed with the covid19 pandemic which also bought a change in the lifestyle of people as they are trying to adjust with the new normal and all this led to give a push to E-Commerce industry as well. Covid19 provided E-Commerce with a new environment so as to expand their footprints with quite a few challenges to overcome like productivity or the supply chain effectiveness. Social distancing gave a positive impact on the E-commerce industries growth. The following Figure shows the revenue earned by the companies in the Quarter 1 of the previous and current year that is 2019 and 2024 has witness an increase even though the economy is facing a downfall.



**FIG: Yearly increasing online purchases**

During the COVID-19 pandemic, online purchasing became increasingly popular as people sought to minimize in-person interactions and adhere to social distancing measures. Here are some key points about online purchasing during this time:

**Increased Demand:** With physical stores closed or operating at reduced capacity, there was a surge in online shopping across various sectors including groceries, household goods, electronics, and more.

**Shift in Consumer Behavior:** Many consumers who had not previously shopped online began to do so out of necessity or convenience during lockdowns and restrictions.

**Safety Concerns:** Online shopping offered a safer alternative to in-store shopping, as it reduced the risk of exposure to the virus.

**Supply Chain Challenges:** The surge in demand placed strain on supply chains, leading to delays in shipping and shortages of certain products.

**Contactless Delivery:** Delivery services implemented contactless delivery options to minimize physical contact between customers and delivery personnel.

**E-commerce Platforms:** Major e-commerce platforms such as Amazon, eBay, and Alibaba experienced increased traffic and sales during this time. Additionally, many local businesses adapted by setting up online stores or partnering with existing platforms.

**Cybersecurity Awareness:** With the rise in online shopping, there was also an increased awareness of cybersecurity risks such as phishing scams and fraudulent websites. Consumers were advised to be vigilant and only purchase from reputable websites.

**Returns and Refunds:** Many retailers adjusted their return and refund policies to accommodate the challenges of online shopping during the pandemic, including extending return windows and offering more flexible refund options.

Overall, online purchasing played a crucial role in enabling consumers to access essential goods and services while minimizing the risk of COVID-19 transmission. Here is a picture that shows the increasing online purchases due to covid19.



**FIG: Increasing online purchases due to covid19**

**SUGGESTIONS**

**✓** Close existing digital divides among individuals, for example by expanding affordable and quality broadband to rural and underserved areas, enhancing financial inclusion, and fostering trust and the acquisition of skills to participate in e-commerce.

**✓** Foster e-commerce participation by the most vulnerable, for example by introducing community-based delivery programmes for elderly and reserved delivery slots. Ensure that vulnerable consumers are protected from unfair business practices and unsafe products.

**✓** Support the creation of innovative e-commerce business models, ensuring that regulatory frameworks remain flexible enough to accommodate combinations of online and offline business functions. Reduce regulatory uncertainty and promote transparency through information sharing.

**✓** Ensure that SMEs can participate in e-commerce, for example by providing policy, regulatory or financial incentives for sales diversification and establishing a level playing for SMEs relying on the services of online platforms.

**✓** Reduce bottlenecks in the enabling environment for e-commerce, including areas such as connectivity, trade, logistics and postal services.

**REPORT CONCLUSION**

E-Commerce industry is that force which cannot be ignored by any element of the economy. Coronavirus pandemic proved that one of the major tools that can help consumers during crisis is e-commerce. In order to maintain social distancing and self quarantine the consumers have become more reliable on the e-commerce industry.

COVID-19 pandemic catalyzed a significant surge in e-commerce activity, with consumers increasingly turning to online shopping for its safety and convenience. This rapid adoption of e-commerce highlighted both opportunities and challenges within the industry. Businesses adapted by enhancing their online presence and implementing measures such as contactless delivery. However, supply chain disruptions and cybersecurity risks posed notable challenges. Despite these obstacles, the pandemic accelerated the digital transformation of retail, with long-term implications expected to shape the e-commerce landscape well beyond the pandemic's end.

**CHAPTER-1**

**INTRODUCTION**

E-commerce is fast gaining ground as an accepted and used business paradigm. More and more business houses are implementing web sites providing functionality for performing commercial transactions over the web. It is reasonable to say that the process of shopping on the web is becoming commonplace. The objective of this project is to develop a generalpurpose e-commerce store where any product (such as books, CDs, computers, mobile phones, electronic items, and home appliances) can be bought from the comfort of home through the Internet. However, for implementation purposes, this paper will deal with an online book store.

An online store is a virtual store on the Internet where customers can browse the catalog and select products of interest. The selected items may be collected in a shopping cart. At checkout time, the items in the shopping cart will be presented as an order. At that time, more information will be needed to complete the transaction. Usually, the customer will be asked to fill or select a billing address, a shipping address, a shipping option, and payment information such as credit card number. An e-mail notification is sent to the customer as soon as the order is placed.

**LITERATURE REVIEW:**

Electronic Commerce (e-commerce) applications support the interaction between different parties participating in a commerce transaction via the network, as well as the management of the data involved in the process [2]. The increasing importance of e-commerce is apparent in the study conducted by researches at the GVU (Graphics, Visualization, and Usability) Center at the Georgia Institute of Technology. In their summary of the findings from the eighth survey, the researchers report that “e-commerce is taking off both in terms of the number of users shopping as well as the total amount people are spending via Internet based transactions”.

Over three quarters of the 10,000 respondents report having purchased items online. The most cited reason for using the web for personal shopping was convenience (65%), followed by availability of vendor information (60%), no pressure form sales person (55%) and saving time (53%). Although the issue of security remains the primary reason why more people do not purchase items online, the GVA survey also indicates that faith in the security of e commerce is increasing. As more people gain confidence in current encryption technologies, more and more users can be expected to frequently purchase items online [11]. A good e-commerce site should present the following factors to the customers for better usability [11]:

• Knowing when an item was saved or not saved in the shopping cart.

• Returning to different parts of the site after adding an item to the shopping cart.

• Easy scanning and selecting items in a list.

• Effective categorical organization of products.

• Simple navigation from home page to information and order links for specific products.

• Obvious shopping links or buttons.

• Minimal and effective security notifications or messages.

• Consistent layout of product information. Another important factor in the design of an e-commerce site is feedback [4]. The interactive cycle between a user and a web site is not complete until the web site responds to a command entered by the user. According to Norman [5], "feedback--sending back to the user information about what action has actually been done, what result has been accomplished--is a wellknown concept in the science of control and information theory. Imagine trying to talk to someone when you cannot even hear your own voice, or trying to draw a picture with a pencil that leaves no mark: there would be no feedback". 2 Web site feedback often consists of a change in the visual or verbal information presented to the user. Simple examples include highlighting a selection made by the user or filling a field on a form based on a user's selection from a pulldown list. Another example is using the sound of a cash register to confirm that a product has been added to an electronic shopping cart. Completed orders should be acknowledged quickly. This may be done with an acknowledgment or fulfillment page. The amount of time it takes to generate and download this page, however, is a source of irritation for many e-commerce users. Users are quick to attribute meaning to events. A blank page, or what a user perceives to be "a long time" to receive an acknowledgment, may be interpreted as "there must be something wrong with the order." If generating an acknowledgment may take longer than what may be reasonably expected by the user, then the design should include intermediate feedback to the user indicating the progress being made toward acknowledgment or fulfillment. Finally, feedback should not distract the user. Actions and reactions made by the web site should be meaningful. Feedback should not draw the user's attention away from the important tasks of gathering information, selecting products, and placing orders.

**1.1 PROJECT AIMS AND OBJECTIVES**

* **Diverse Book Selection:** Offer a wide range of books across genres and formats to cater to diverse interests.
* **Intuitive Browsing:** Provide a user-friendly interface for easy book discovery and browsing.
* **Efficient Inventory Management:** Implement systems to manage book inventory effectively, ensuring accurate stock levels.
* **Secure Checkout Process:** Ensure a secure and seamless checkout experience with multiple payment options.
* **Personalized Recommendations:** Utilize algorithms to offer personalized book suggestions based on user preferences.
* **Mobile Optimization:** Optimize the platform for mobile devices for convenient browsing and purchasing.
* **Community Engagement:** Foster a sense of community among users through book clubs and discussion forums.
* **Efficient Order Fulfillment:** Implement streamlined processes for timely delivery of orders.
* **Responsive Customer Support:** Offer responsive support channels to address customer inquiries and feedback.
* **Promotional Campaigns:** Run promotions and discounts to attract customers and encourage repeat purchases.
* **Content Expansion:** Expand offerings beyond books to include audiobooks and related merchandise.
* **Partnerships with Publishers:** Establish partnerships for access to a wide range of titles and timely availability.
* **Data Analytics:** Utilize data analytics to gain insights for marketing and inventory management.
* **Accessibility:** Ensure accessibility for all users, including those with disabilities.
* **Inclusivity:** Offer content in multiple languages and formats to cater to diverse audiences.

**1.2 BACKGROUND OF PROJECT**

The project aims to tap into the growing trend of online shopping, particularly in the book industry. With the increasing digitalization of content consumption and the convenience of online shopping, there is a significant opportunity to provide readers with a platform that offers a diverse range of books accessible from anywhere, at any time. By leveraging technology and user-friendly interfaces, the project seeks to create a seamless online book-buying experience, catering to modern consumers' preferences for convenience, choice, and personalized recommendations. the needs that are used in the ecommerce book industry is shown below

|  |  |
| --- | --- |
| **Component** | **Description** |
| Processor | Intel Core i7 or AMD Ryzen 7 processor (or equivalent) |
| Operating Systems | Linux (Ubuntu, CentOS) or Windows Server |
| Memory | Minimum 8GB RAM, recommended 16GB RAM |
| Hard Disk Space | SSD storage recommended for faster performance |
| Data Base | MySQL or PostgreSQL for relational database management |
| Web Server | Apache HTTP Server or Nginx |
| Programming Language | Backend: Node.js, Python, or Java |
| Frontend Framework | React.js, Angular, or Vue.js |
| Payment Gateway | Integration with Stripe, PayPal, or other payment services |

**CHAPTER-2**

**SYSTEM ANALYSIS**

This chapter provides a comprehensive overview of the development process, including the software requirement specification and a comparative analysis between the existing and proposed systems for the online book sales e-commerce platform.

**Chapter Title:** Development Process of the Online Book Sales E-Commerce Platform

**1. Introduction**

Brief overview of the project scope and objectives.

**2. Software Requirement Specification (SRS)**

Definition and importance of SRS.

Functional requirements:

User functionalities (e.g., browsing books, adding to cart, checkout).

Admin functionalities (e.g., managing inventory, processing orders).

Non-functional requirements:

Performance (e.g., page load times, server responsiveness).

Security (e.g., encryption, secure authentication).

Usability (e.g., intuitive user interfaces, responsiveness).

Scalability (e.g., handling increasing user traffic).

Compatibility (e.g., cross-browser compatibility, mobile responsiveness).

**3. Comparison between Existing and Proposed Systems**

Overview of the existing system (if applicable):

Key features and functionalities.

Limitations and challenges.

Proposed enhancements and improvements:

Introduction of the online book sales platform.

Detailed comparison highlighting the advantages of the proposed system over the existing one:

Improved user experience: User-friendly interfaces, personalized recommendations.

Expanded functionalities: Enhanced search and filtering options, streamlined checkout process.

Scalability and performance: Ability to handle increased traffic and transactions.

Security enhancements: Implementation of robust security measures.

Technological advancements: Integration with modern technologies and frameworks.

Potential impact of the proposed system on stakeholders:

Improved efficiency and productivity for administrators.

Enhanced shopping experience and satisfaction for users.

Potential for increased sales and revenue for the business.

**4. Conclusion**

Summary of key points discussed in the chapter.

Importance of the development process and SRS in ensuring the success of the project.

Anticipated benefits of the proposed system for stakeholders.

**2.1 SOFTWARE REQUIREMENT SPECIFICATION**

**2.1.1 GENERAL DESCRIPTION**

**PRODUCT DESCRIPTION**

The Online Book Sales Platform is a digital system designed to facilitate the management of daily activities related to selling books in an electronic format. It serves as a centralized platform for users (administrators) to efficiently handle transactions and records, mitigating risks associated with traditional paperwork such as file loss and damage, while significantly reducing time consumption.

**PROBLEM STATEMENT**

Prior to the implementation of a computerized system, the following issues arise:

**Risk of File Loss:**

Without a computerized system, there's a constant risk of losing files due to the human environment. Human errors can lead to the misplacement or loss of important records.

**File Damage:**

In the absence of a computerized system, files are susceptible to damage from accidents such as spillage or natural disasters like floods or fires, leading to loss of data.

**Difficulty in Record Retrieval:**

Managing large volumes of records without a computerized system makes it challenging to search and retrieve specific records efficiently.

**Space Consumption:**

As the number of records grows, physical storage space required for files and records increases significantly, leading to space constraints within the library premises.

**Cost Inefficiency:**

Without a computerized system, each record addition necessitates the use of paper, resulting in increased operational costs for the library management.

These challenges highlight the urgent need for the implementation of a computerized system to streamline operations and mitigate the risks associated with manual record keeping process.

**2.1.2 SYSTEM OBJECTIVES**

Improvement in Control and Performance:

Develop the e-commerce platform to effectively address current challenges and limitations in online book sales management. Ensure the system is capable of adding users, validating user inputs, and maintaining high performance standards without bugs.

**Cost Savings:**

Implementing the online book sales platform aims to reduce the need for manual intervention, leading to decreased operational costs associated with managing physical bookstores. This includes savings on staffing, inventory management, and infrastructure maintenance.

**Time Efficiency:**

Enable users to swiftly browse and purchase books through the platform, saving time compared to traditional in-store shopping experiences. Provide efficient search functionalities and intuitive user interfaces to streamline the book discovery and checkout process.

**Option of Online Notice Board:**

Introduce features such as an online notice board to inform users about upcoming book launches, author events, and promotional offers. This enhances user engagement and fosters a sense of community within the online book sales platform.

**Enhanced Access to Educational Resources:**

Provide users with access to a wide range of educational resources, including lecture notes and study materials. Enable educators to upload and share educational content in various formats, enriching the learning experience for students.

These objectives aim to address the specific challenges and requirements of the online book sales e-commerce platform, focusing on improving control, efficiency, and user experience while leveraging technology to enhance the management and dissemination of educational resources.

**2.1.3 SYSTEM REQUIREMENTS**

**2.1.3.1 Non-Functional Requirements**

**Product Requirements**

**Efficiency Requirement:**

Upon implementation of the e-commerce platform, users (both customers and administrators) should experience significantly faster access to the online bookstore. Processes such as searching for books and completing transactions should be swift and efficient.

**Reliability Requirement:**

The system must reliably perform key functionalities such as member registration, member validation, report generation, book transactions, and search operations. Accuracy and consistency in executing these tasks are essential to ensure user satisfaction.

**Usability Requirement:**

The e-commerce platform should offer a user-friendly environment, enabling students, staff, and customers to navigate and perform tasks easily and effectively. Intuitive interfaces and clear instructions should enhance usability for all users.

**Organizational Requirements**

**Implementation Requirements:**

The front-end of the system will be developed using HTML, with PHP serving as the server-side scripting language for database connectivity. The backend, which encompasses the database, will be built using MySQL. This tech stack ensures robust functionality and seamless integration across system components.

**Delivery Requirements:**

The entire e-commerce platform project is expected to be delivered within a timeframe of six months. Regular evaluations by the project guide will be conducted on a weekly basis to monitor progress and ensure adherence to project timelines.

These non-functional requirements set the foundation for the development and implementation of the comprehensive e-commerce platform for online book sales, focusing on efficiency, reliability, usability, and organizational aspects to deliver a high-quality solution within the specified timeframe.

**2.1.3.2 FUNCTIONAL REQUIREMENTS**

**1. User Management:**

**1.1 User Registration:**

Users shall be able to register for an account by providing necessary information, including name, email address, and password.

The system shall verify the uniqueness of the email address during registration.

Upon successful registration, users shall receive a confirmation email.

**1.2 User Authentication:**

Registered users shall be able to log in using their email address and password.

The system shall authenticate users' credentials before granting access to account features.

**2. Product Management:**

**2.1 Add New Books:**

Admin users shall have the ability to add new books to the platform.

The system shall require mandatory fields such as book title, author, ISBN, description, price, and quantity.

Duplicate ISBNs shall not be allowed for new book entries.

**2.2 Search and Filter Books:**

Users shall be able to search for books based on various criteria such as title, author, genre, ISBN, etc.

The system shall provide filters for refining search results by category, price range, rating, etc.

Search results shall be displayed in a user-friendly manner, with options for sorting.

**3. Shopping Cart and Checkout:**

**3.1 Add to Cart:**

Users shall be able to add books to their shopping cart from the product listing or book details page.

The system shall update the cart dynamically to reflect added items and total cost.

**3.2 Modify Cart:**

Users shall have the ability to edit the quantity or remove items from their shopping cart.

Changes to the cart shall be reflected in real-time, updating the total cost accordingly.

**3.3 Checkout Process:**

Users shall be guided through a secure checkout process to complete their purchase.

The system shall require shipping and billing information from users.

Users shall have the option to choose from available payment methods (credit/debit card, PayPal, etc.).

Upon successful completion of the checkout process, users shall receive a confirmation email with order details.

**4. Order Management:**

**4.1 View Order History:**

Registered users shall be able to view their order history, including past purchases, order status, and tracking information.

**2.1.4 SOFTWARE AND HARDWARE REQUIREMENTS**

**Software Requirements:**

**Operating System:**

Linux, preferably Ubuntu Server or CentOS, for hosting the web server.

**Web Server:**

Apache or Nginx to serve web pages to users.

**Database Management System**:

MySQL or PostgreSQL for storing product information, user data, and order details.

**Programming Languages and Frameworks:**

**Frontend:**

HTML, CSS, JavaScript (ES6) for building user interfaces.

Frontend Frameworks like React.js or Vue.js for dynamic web application development.

**Backend:**

Node.js, Python (with Django or Flask), or Ruby (with Ruby on Rails) for server-side logic.

**Database:**

Sequelize (for Node.js), Django ORM (for Python), or ActiveRecord (for Ruby) for interacting with the database.

**Payment Gateway Integration:**

Integration with payment gateways like PayPal, Stripe, or Square for processing online payments securely.

**Security Measures:**

Implementation of SSL/TLS certificates for secure data transmission (HTTPS).

User authentication and authorization mechanisms.

Input validation and sanitization to prevent common security vulnerabilities like SQL injection and cross-site scripting (XSS).

**Content Delivery Network (CDN):**

Integration with CDN services like Cloudflare or AWS CloudFront for faster content delivery and DDoS protection.

**Version Control:**

Git for version control and collaboration among developers.

**Hardware Requirements:**

**Server Hardware:**

Virtual Private Server (VPS) or Dedicated Server with sufficient computing resources:

**CPU:** At least a dual-core processor (4 cores or more recommended for high traffic).

**RAM:** Minimum 2 GB (4 GB or more recommended).

**Storage:** SSD storage for faster data access.

Alternatively, cloud-based services like AWS EC2, Google Cloud Compute Engine, or Microsoft Azure VMs can be used.

**Database Server:**

Separate server for hosting the database, especially for larger-scale applications.

Similar specifications as the web server but with higher RAM allocation depending on the database size and traffic.

**Backup Solutions:**

Regular backups of the database and website files to ensure data integrity and disaster recovery.

**Load Balancer (Optional):**

For distributing incoming traffic across multiple servers to improve performance and reliability, especially for high-traffic websites.

**Monitoring Tools:**

Monitoring software like Prometheus, Grafana, or New Relic for tracking server performance, uptime, and application health.

**Scalability Considerations:**

Architecture designed for horizontal scalability to accommodate increasing traffic and user demand.

Ensure that the chosen hardware and software components meet the scalability, security, and performance requirements of the e-commerce platform for online book sales. Regular maintenance, updates, and security audits are essential for maintaining the system's integrity and protecting user data.

This section describes the software and hardware requirements of the system.

..**Online registration for students.**

Individual member accounts for accessing information.

Comprehensive book details management, including authors, number of copies, availability status, etc.

Management of issued books, return dates, and fines for delays.

Administrator privileges for adding and updating books.

Reduced time consumption, improved accuracy, and reliability.

**2.3 SOFTWARE TOOLS USED**

The whole project is divided into two parts. The frontend and the backend.

**2.3.1 FRONTEND**

The frontend is designed is using of HTML, CSS, JAVASCRIPT

**Front End Development:**

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

Use HTML to structure the layout of your web pages, including headers, footers, navigation bars, and content sections.

Create HTML forms for user input, such as search forms, login/registration forms, and checkout forms.

Embed HTML elements to display book listings, product details, and promotional banners.

**CSS (Cascading Style Sheets):**

Apply CSS styles to enhance the visual appearance of your e-commerce platform, including fonts, colors, margins, and padding.

Implement responsive design techniques to ensure your website is accessible and user-friendly on various devices and screen sizes.

Use CSS animations and transitions to add interactivity and improve user engagement.

**JavaScript:**

Enhance user experience with client-side interactivity using JavaScript.

Implement features such as product filtering, live search, image sliders, and dynamic content loading to improve usability.

Validate user input on forms to ensure data integrity and provide real-time feedback.

**Back End Development:**

**PHP (Hypertext Preprocessor):**

Develop server-side scripts in PHP to handle user authentication, session management, and data processing.

Create PHP scripts to interact with the MySQL database for tasks such as retrieving book information, processing orders, and managing user accounts.

Implement security measures, such as input validation, SQL injection prevention, and authentication mechanisms, to protect against vulnerabilities.

**MySQL (Structured Query Language):**

Design a relational database schema to store information about books, users, orders, and other relevant data entities.

Use SQL queries to perform CRUD operations (Create, Read, Update, Delete) on the database, including inserting new records, fetching data, updating existing entries, and deleting obsolete data.

Optimize database performance through indexing, normalization, and query optimization techniques to ensure efficient data retrieval and processing.

**Full Stack Integration:**

Integrate front-end and back-end components seamlessly to create a cohesive user experience.

Implement RESTful APIs or server-side endpoints to facilitate communication between the client-side and server-side components.

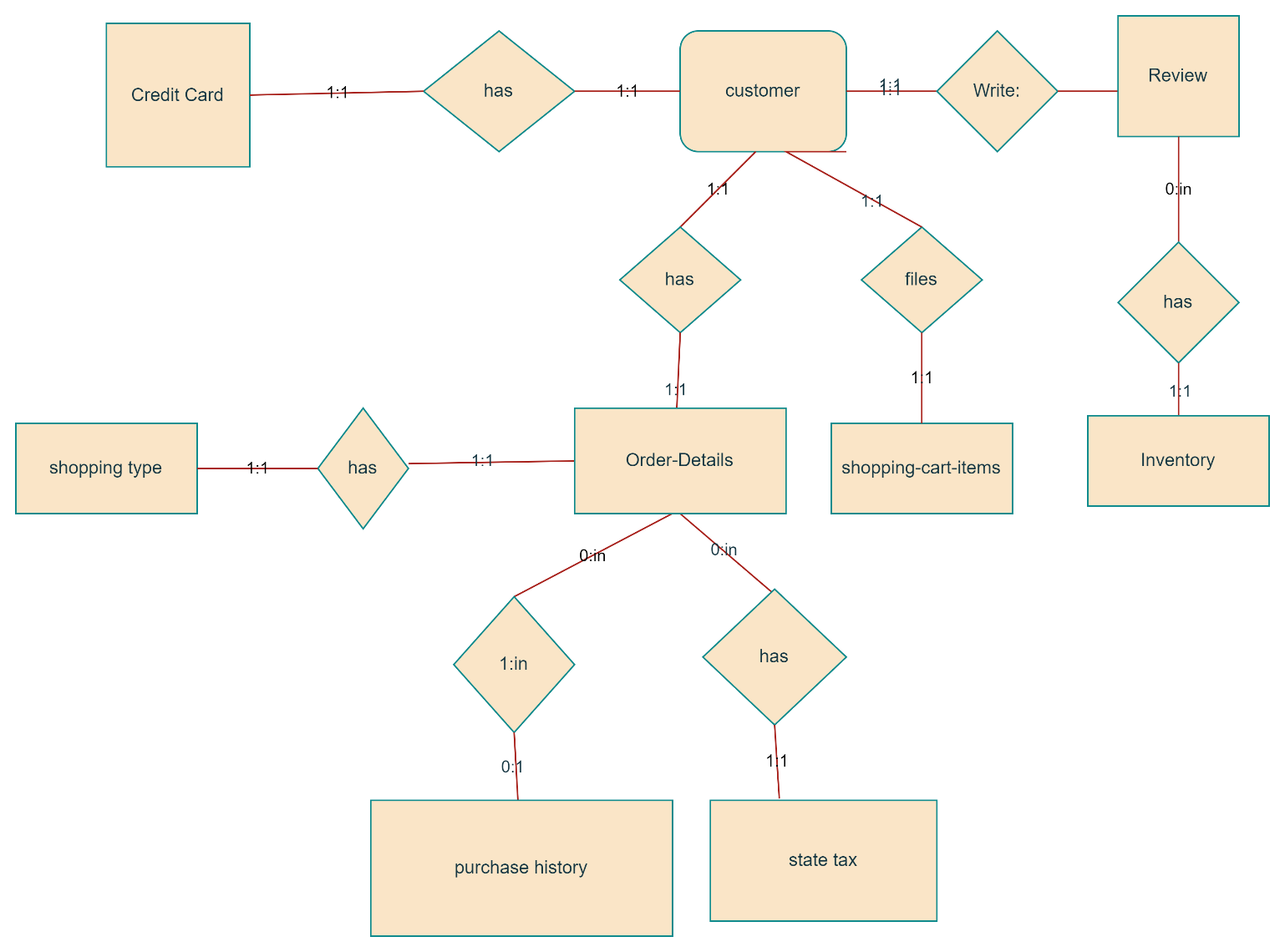
Ensure data consistency and synchronization between the client and server to maintain a reliable e-commerce platform.

Test the application thoroughly to identify and resolve any bugs or issues, ensuring a smooth and error-free user experience.

**CHAPTER-3**

**PROJECT DESIGN**

In order to design a web site, the relational database must be designed first. Conceptual design can be divided into two parts: The data model and the process model. The data model focuses on what data should be stored in the database while the process model deals with how the data is processed. To put this in the context of the relational database, the data model is used to design the relational tables. The process model is used to design the queries that will access and perform operations on those tables.



**FIGURE:1 ENTITY RELATIONSHIP DIAGRAM**

entity A matches exactly one record in entity B and every record in B matches exactly one record in A. One to many means that every record in A matches zero or more records in B and every record in B matches exactly one record in A. If there is a one to many relationship between two entities, then these entities are represented as Associative Entities. In the Relational Database model, each of the entities will be transformed into a table. The tables are shown below along with the attributes.

**3.1.1 Database Design**

In this section, the basic structure of the tables composing the database for the project are shown along with information about primary and foreign keys.

**Customer**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **NAME** | **TYPE** | **DESCRIPTION** |
| 1 | User Id | Varchar | Primary key for customer identification |
| 2 | Password | Varchar | Security for customer |
| 3 | First\_name | Varchar |  |
| 4 | Last\_name | Varchar |  |
| 5 | Address | Varchar |  |
| 6 | City | Varchar |  |
| 7 | Zip | Integer |  |
| 8 | State | varchar |  |
| 9 | Email Address | Varchar |  |
| 10 | Phone\_Number | Varchar |  |

**Books**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **NAME** | **TYPE** | **DESCRIPTION** |
| 1 | Inventory\_ID | Integer | Primary key for inventory identification, ISBN of a book |
| 2 | Book\_Name | Varchar |  |
| 3 | Author | Varchar |  |
| 4 | Nr\_books | Integer |  |
| 5 | Price | Double |  |

**State\_Tax**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **NAME** | **TYPE** | **DESCRIPTION** |
| 1 | State Name | Varchar | Primary key for state identification |
| 2 | Sales Tax Rate | Double | Sales Tax for each state Shopping\_Cart\_Items |

**Shopping\_cart\_items**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **NAME** | **TYPE** | **DESCRIPTION** |
| 1 | Shopping cart\_ID | Integer | Primary key for shopping cart identification |
| 2 | Inventory ID | Varchar | Foreign key to inventory |
| 3 | Price | Double |  |
| 4 | Date | Date |  |
| 5 | User ID | Varchar | Forign key to customer |
| 6 | Quantity | Integer |  |

**Order\_Details**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **NAME** | **TYPE** | **DESCRIPTION** |
| 1 | OrderID  UserID | Integer  Char | Primary key for order identification  Foreign key to customer |
| 2 | Reciever’s Name | Char | If order is to be sent to other address rather than to the customer, we need that address |
| 3 | Address | Char |  |
| 4 | City | Char |  |
| 5 | Zip | Integer |  |
| 6 | State | Char | Foreign key to State Tax |
| 7 | type of shipping | Char | Foreign key to Shipping Type |
| 8 | Date of purchase | Date |  |

**Shipping\_Type**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **NAME** | **TYPE** | **DESCRIPTION** |
| 1 | Type of Shipping | Varchar | Primary key to define type of shipping |
| 2 | Price | Double |  |
| 3 | Approximate days for delivery | Integer |  |

**Credit\_Card\_Details**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **NAME** | **TYPE** | **DESCRIPTION** |
| 1 | credit user name | Varchar | Primary key for customer identification |
| 2 | Credit card number | varchar |  |
| 3 | Card type | Varchar | Master card, visa, discover |
| 4 | CVV number | Integer | Number present on the back of the card for extra security |
| 5 | Expiry Date | Date |  |
| 6 | UserID | Varchar | Foreign key to Customer |

**Book\_Review**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **NAME** | **TYPE** | **DESCRIPTION** |
| 1 | InventoryID | Varchar | ISBN of the book on which the review is written |
| 2 | Reviews | Varchar | Review on the book |
| 3 | Rating | Varchar | Rating given to the book in a scale of 5 4 Review Date Date |
| 4 | Reeview Date | Date |  |
| 5 | User Name | varchar | Name of the user providing the review |

**Purchase\_History**

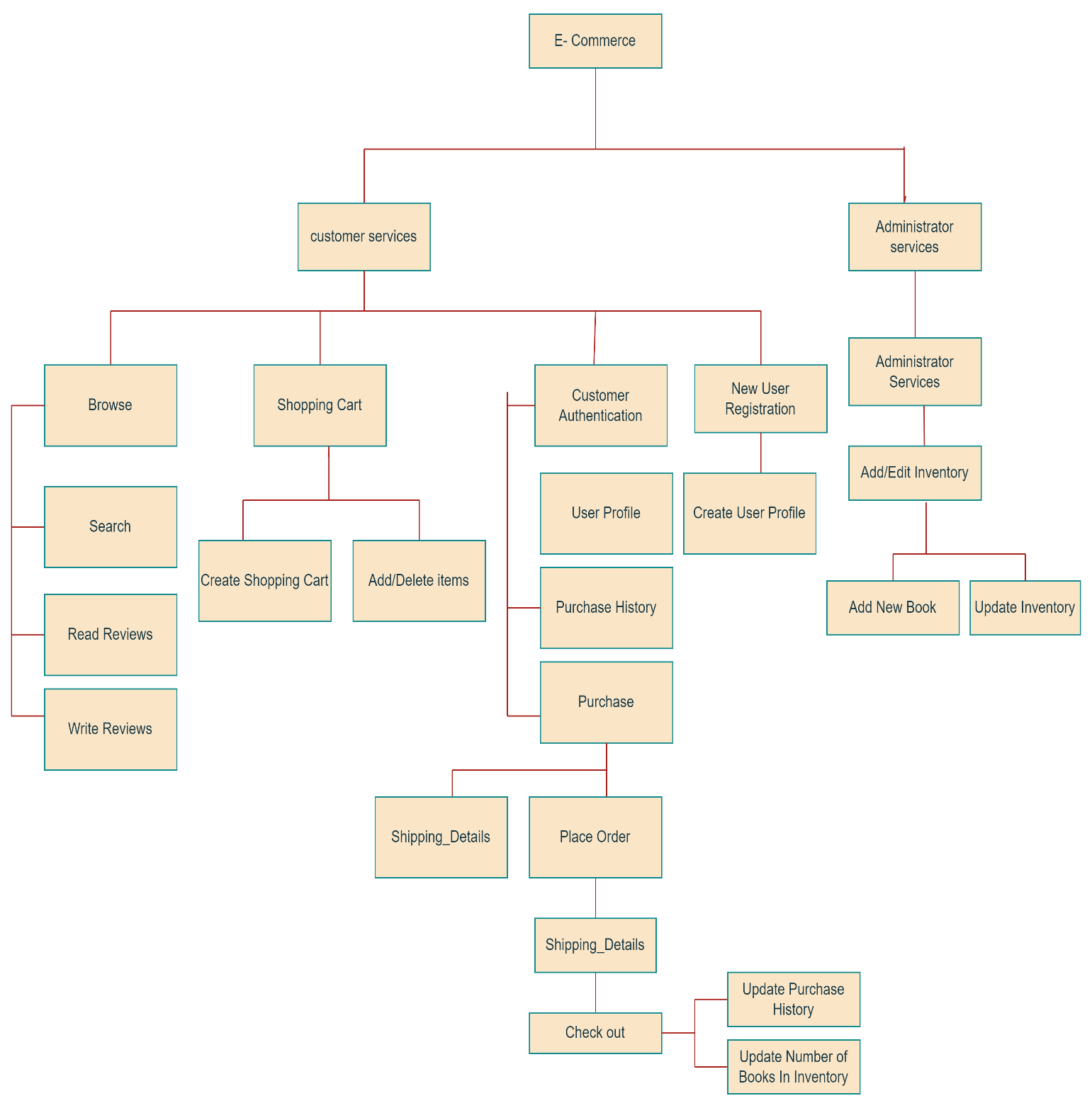
|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **NAME** | **TYPE** | **DESCRIPTION** |
| 1 | UserID | Varchar | Primary key for Customer Identification |
| 2 | InveentoryID | varchar | Book purchased by the user |
| 3 | Date of purchase | Date |  |
| 4 | OrderID | Integer | Foreign key to Order\_details |
| 5 | Quantity | Integer |  |
| 6 | Price double |  |  |

**3.2. Process Model**

A Process Model tells us about how the data is processed and how the data flows from one table to another to gather the required information. This model consists of the Functional Decomposition Diagram and Data Flow Diagram.

**3.2.1. Functional Decomposition Diagram**

A decomposition diagram shows a top-down functional decomposition of a system and exposes the system's structure. The objective of the Functional Decomposition is to break down a system step by step, beginning with the main function of a system and continuing with the interim levels down to the level of elementary functions. The diagram is the starting point for more detailed process diagrams, such as data flow diagrams (DFD). Figure 2 shows the Functional Decomposition Diagram for this project.



**FIGURE 2-FUNCTIONAL DECOMPOSITION DIAGRAM**

**3.2.2 Data Flow Diagram (DFD)**

Data Flow Diagrams show the flow of data from external entities into the system, and from one process to another within the system. There are four symbols for drawing a DFD:

1. Rectangles representing external entities, which are sources or destinations of data.

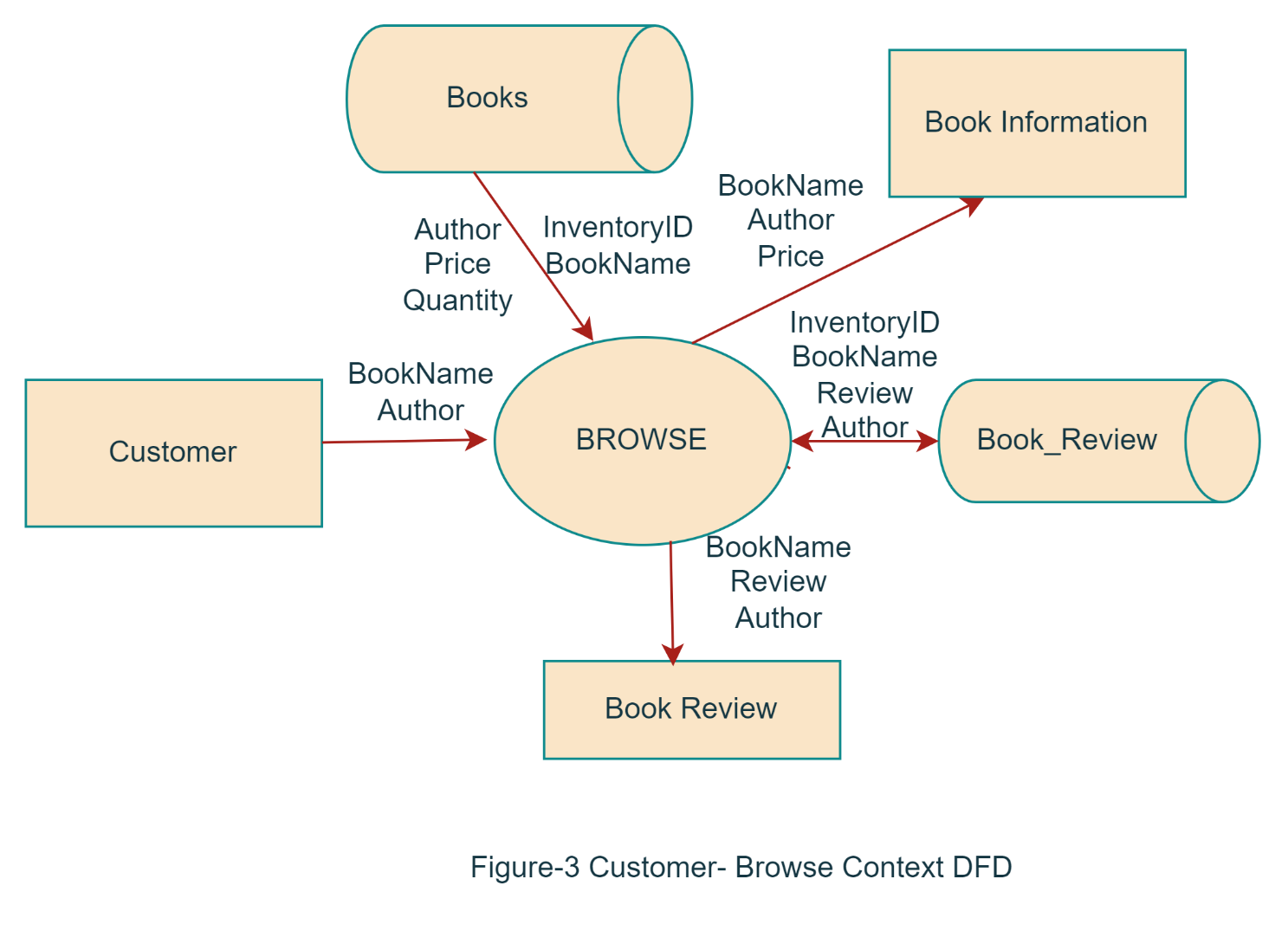
2. Ellipses representing processes, which take data as input, validate and process it and output it.

3. Arrows representing the data flows, which can either, be electronic data or physical items.

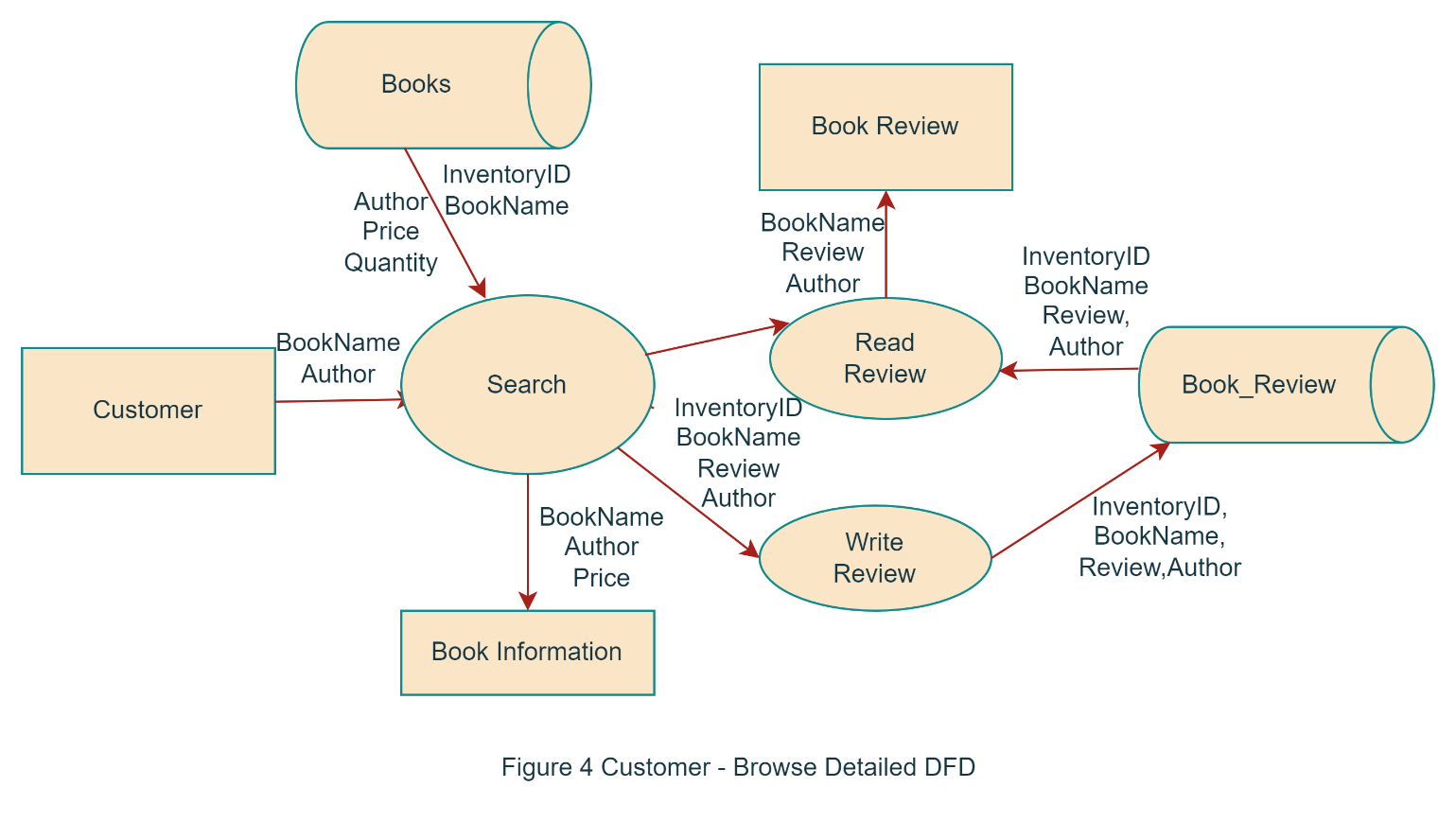
4. Open-ended rectangles or a Disk symbol representing data stores, including electronic stores such as databases or XML files and physical stores such as filing cabinets or stacks of paper.

Figures 3 - 14 are the Data Flow Diagrams for the current system. Each process within the system is first shown as a Context Level DFD and later as a Detailed DFD. The Context Level DFD provides a conceptual view of the process and its surrounding input, output and data stores. The Detailed DFD provides a more detailed and comprehensive view of the interaction among the sub-processes within the system.

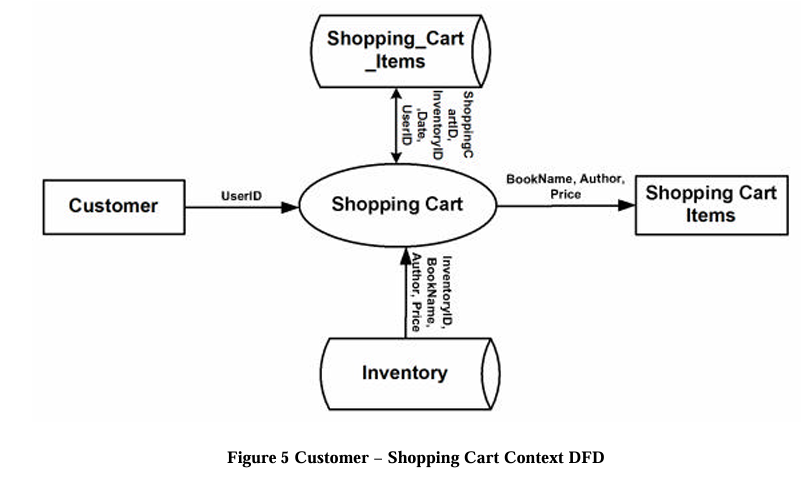
**Customer- Browse Context DFD**



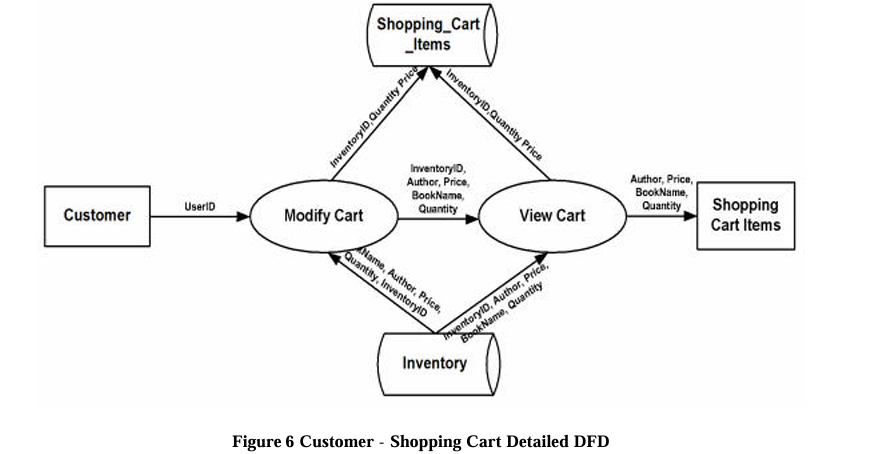
**Customer-Browse Detailed DFD**



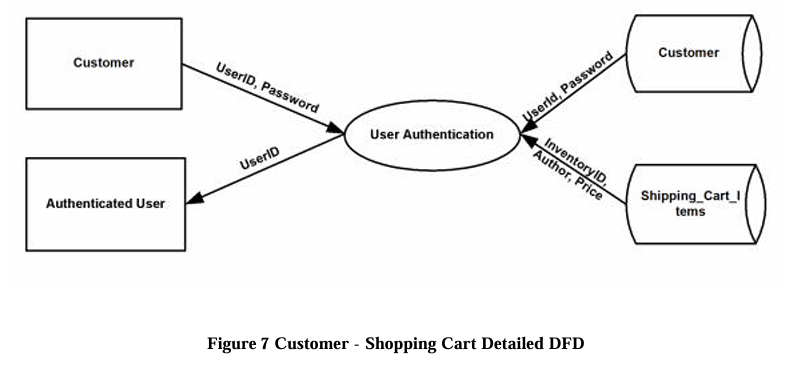
**Customer- Shopping Cart Context DFD**



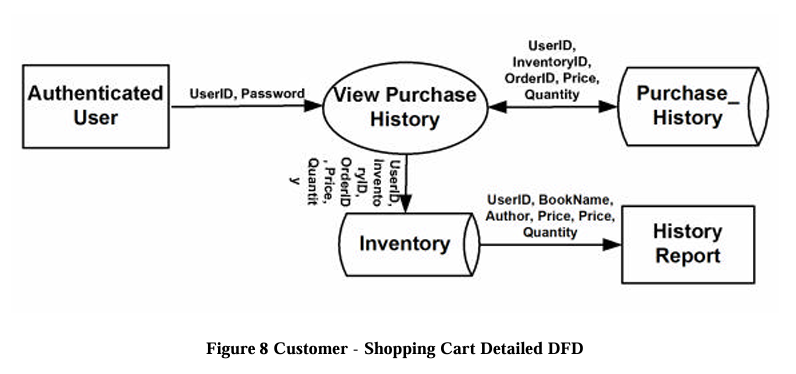
**Customer - ShoppingCart Detailded DFD**

****

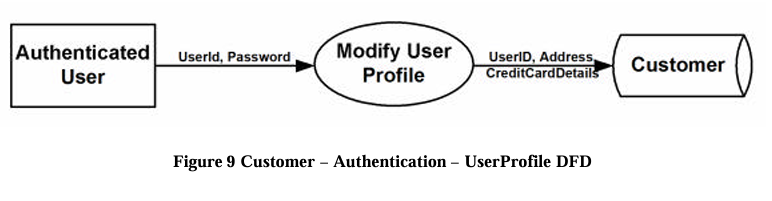
**Customer-Authentication Context DFD**



**Customer-Authentication-PurchaseHistory DFD**

****

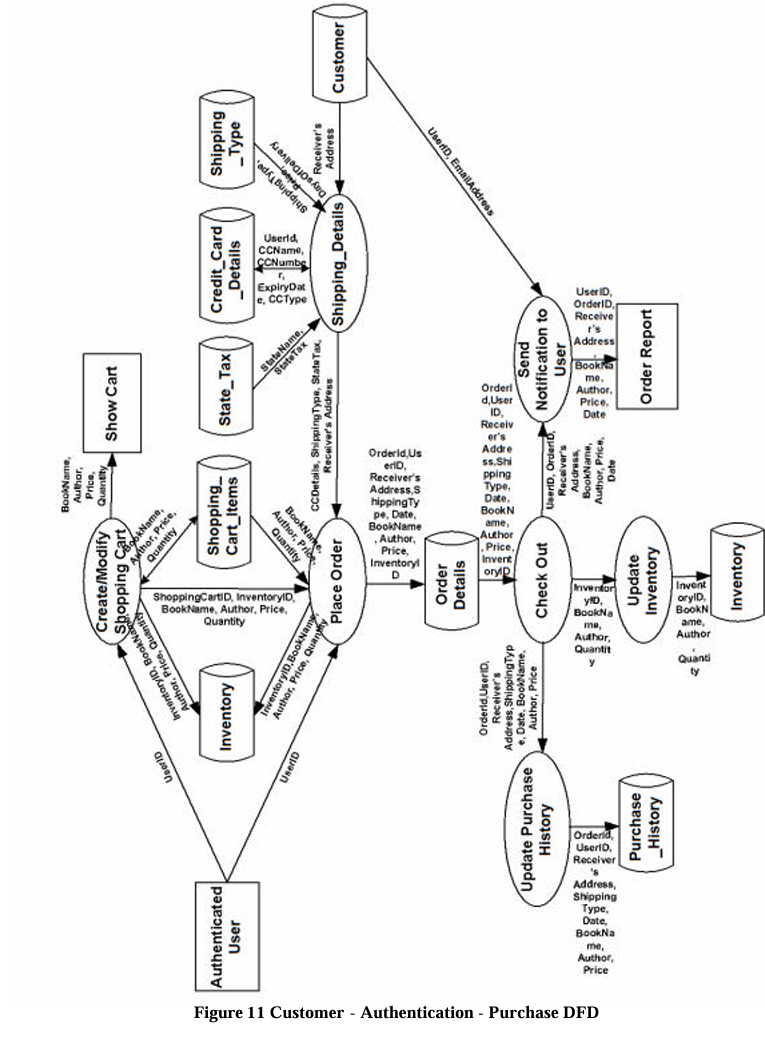
**Customer-Authentication-UserProfile DFD**

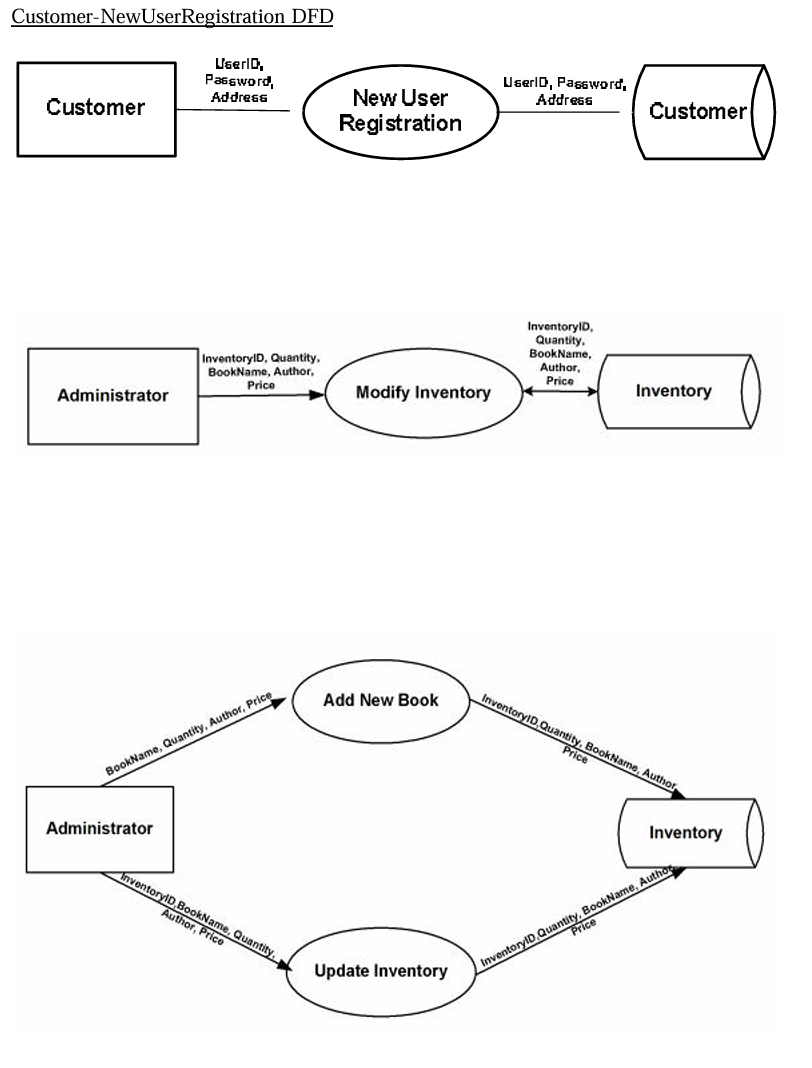


**Authenticated User-Purchase Context DFD**

****

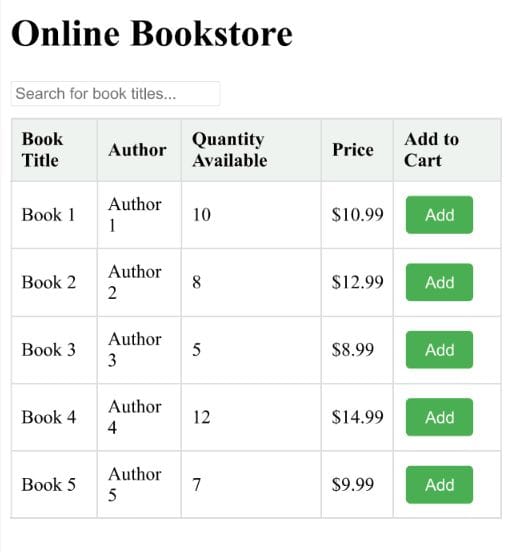
**Authenticated User-Purchase DFD**



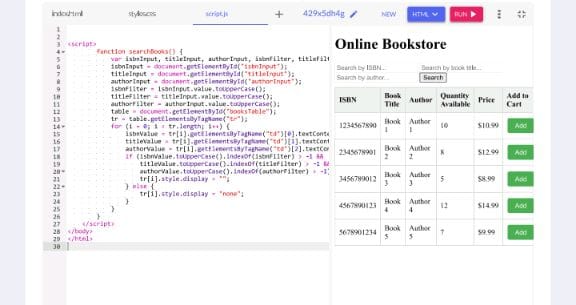
****

**3.3 User Interface Design**

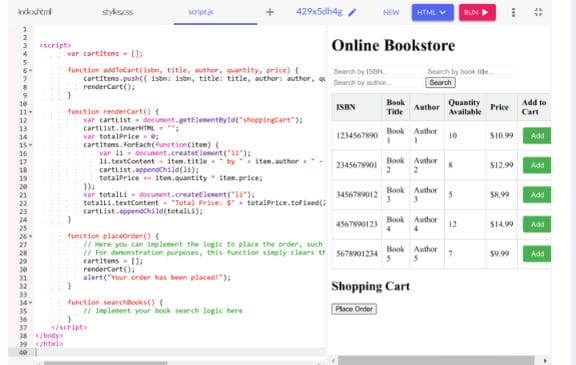
Before implementing the actual design of the project, a few user interface designs were constructed to visualize the user interaction with the system as they browse for books, create a shopping cart and purchase books. The user interface design will closely follow our Functional Decomposition Diagram (Figure 2). Figures 15 – 20 show the initial designs of the web pages.

****

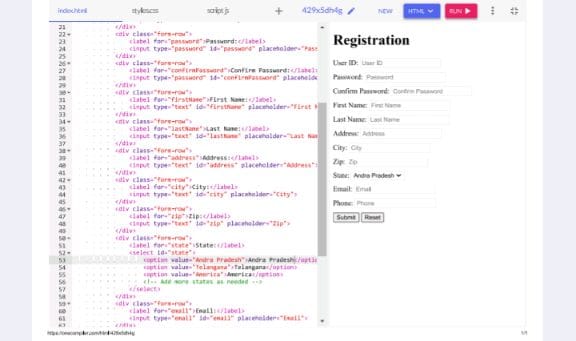
**FIG:15 MENU AND DISPLAY OF BOOKS IN THE STORE**

****

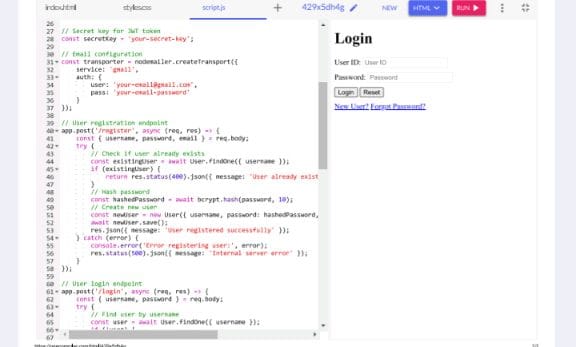
**FIG:16 FOR SEARCHING THE BOOKS IN STORE**

****

**FIG:17 SHOPPING CART FOR THE USER**

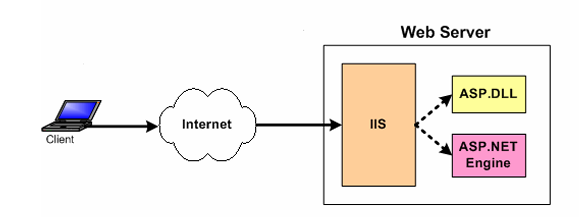
****

**FIG:18 REGISTRATION OF THE NEW USER**

****

**FIG:19 AUTHENTICATION OF THE USER**

**4.IMPLEMENTATION TECHNOLOGIES**

****The objective of this project is to develop an online book store. When the user types in the URL of the Book Store in the address field of the browser, a Web Server is contacted to get the requested information. In the .NET Framework, IIS (Internet Information Service) acts as the Web Server. The sole task of a Web Server is to accept incoming HTTP requests and to return the requested resource in an HTTP response. The first thing IIS does when a request comes in is to decide how to handle the request. Its decision is based upon the requested file's extension. For example, if the requested file has the .asp extension, IIS will route the request to be handled by asp.dll. If it has the extension of .aspx, .ascx, etc, it will route the request to be handled by ASP.NET Engine.

**FIGURE:20 RELATION BETWEEN IIS AND ASP.NET**

The ASP.NET Engine then gets the requested file, and if necessary contacts the database through ADO.NET for the required file and then the information is sent back to the Client’s browser. Figure 20 shows how a client browser interacts with the Web server and how the Web server handles the request from client.

**4.1 INTERNET INFORMATION SERVICES(IIS)**

IIS is a set of Internet based services for Windows machines. Originally supplied as part of the Option Pack for Windows NT, they were subsequently integrated with Windows 2000 and Windows Server 2003). The current (Windows 2003) version is IIS 6.0 and includes servers for FTP (a software standard for transferring computer files between machines with widely different operating systems), SMTP (Simple Mail Transfer Protocol, is the de facto standard for email transmission across the Internet) and HTTP/HTTPS (is the secure version of HTTP, the communication protocol of the World Wide Web) [12].

**Features:** The web server itself cannot directly perform server side processing but can delegate the task to ISAPI (Application Programming Interface of IIS) applications on the server. Microsoft provides a number of these including ones for Active Server Page and ASP.NET.

**Compatibility:** Internet Information Services is designed to run on Windows server operating systems. A restricted version that supports one web site and a limited number of connections is also supplied with Windows XP Professional. Microsoft has also changed the server account that IIS runs on. In versions of IIS before 6.0, all the features were run on the System account, allowing exploits to run wild on the system. Under 6.0 many of the processes have been brought under a Network Services account that has fewer privileges. In particular this means that if there were an exploit on that feature, it would not necessarily compromise the entire system.

**4.2 ASP.NET**

ASP.NET is a programming framework built on the common language runtime that can be used on a server to build powerful Web applications. ASP.NET has many advantages – both for programmers and for the end users because it is compatible with the .NET Framework. This compatibility allows the users to use the following features through ASP.NET:

**a) Powerful database-driven functionality:** ASP.NET allows programmers to develop web applications that interface with a database. The advantage of ASP.NET is that it is object-oriented and has many programming tools that allow for faster development and more functionality.

**b) Faster web applications:** Two aspects of ASP.NET make it fast -- compiled code and caching. In ASP.NET the code is compiled into "machine language" before a visitor ever comes to the website. Caching is the storage of information in memory for faster access in the future. ASP.NET allows programmers to set up pages or areas of pages that are commonly reused to be cached for a set period of time to improve the performance of web applications. In addition, ASP.NET allows the caching of data from a database so the website is not slowed down by frequent visits to a database when the data does not change very often.

**c) Memory leak and crash protection:** ASP.NET automatically recovers from memory leaks and errors to make sure that the website is always available to the visitors. ASP.NET also supports code written in more than 25 .NET languages (including VB.NET, C#, and Jscript.Net). This is achieved by the Common Language Runtime (CLR) compiler that supports multiple languages.

**4.2.1. Authentication in ASP**

NET There are two separate authentication layers in an ASP.NET application. All requests flow through IIS before they are handed to ASP.NET, and IIS can decide to deny access before ASP.NET even knows about the request. Here is how the process works [14]:

1. IIS checks to see if an incoming request is coming from an IP address that is allowed access to the domain. If not, the request is denied.

2. IIS performs its own user authentication, if it is configured to do so. By default, IIS allows anonymous access and requests are authenticated automatically.

3. When a request is passed from IIS to ASP.NET with an authenticated user, ASP.NET checks to see whether impersonation is enabled. If so, ASP.NET acts as though it were the authenticated user. If not, ASP.NET acts with its own configured account.

4. Finally, the identity is used to request resources from the operating system. If all the necessary resources can be obtained, the user's request is granted; other wise the request is denied.

**4.3. MySQL Database**

In this project, MySQL is used as the backend database. MySQL is an opensource database management system. The features of MySQL are given below:

• MySQL is a relational database management system. A relational database stores information in different tables, rather than in one giant table. These tables can be referenced to each other, to access and maintain data easily.

• MySQL is opensource database system. The database software can be used and modify by anyone according to their needs.

• It is fast, reliable and easy to use. To improve the performance, MySQL is multi threaded database engine. A multithreaded application performs many tasks at the same time as if multiple instances of that application were running simultaneously. In being multithreaded MySQL has many advantages. A separate thread handles each incoming connection with an extra thread that is always running to manage the connections. Multiple clients can perform read operations simultaneously, but while writing, only hold up another client that needs access to the data being updated. Even though the threads share the same process space, they execute individually and because of this separation, multiprocessor machines can spread the thread across many CPUs as long as the host operating system supports multiple CPUs. Multithreading is the key feature to support MySQL’s performance design goals. It is the core feature around which MySQL is built. MySQL database is connected to ASP.NET using an ODBC driver. Open Database Connectivity (ODBC) is a widely accepted application-programming interface (API) for database access. The ODBC driver is a library that implements the functions supported by ODBC API. It processes ODBC function calls, submits SQL requests to MySQL server, and returns results back to the application. If necessary, the driver modifies an application's request so that the request conforms to syntax supported by MySQL.

**4.4. Integrating IIS and ASP.NET**

When a request comes into IIS Web server its extension is examined and, based on this extension, the request is either handled directly by IIS or routed to an ISAPI extension. An ISAPI extension is a compiled class that is installed on the Web server and whose responsibility is to return the markup for the requested file type. By default, IIS handles the request, and simply returns the contents of the requested file [13]. This makes sense for static files, like images, HTML pages, CSS files, external JavaScript files, and so on. For example, when a request is made for a .html file, IIS simply returns the contents of the requested HTML file.

For files whose content is dynamically generated, the ISAPI extension configured for the file extension is responsible for generating the content for the requested file. For example, a Web site that serves up classic ASP pages has the .asp extension mapped to the asp.dll ISAPI extension.

The asp.dll ISAPI extension executes the requested ASP page and returns its generated HTML markup. If the Web site serves up ASP.NET Web pages, IIS has mapped the .aspx to aspnet\_isapi.dll, an ISAPI extension that starts off the process of generating the rendered HTML for the requested ASP.NET Web page. The aspnet\_isapi.dll ISAPI extension is a piece of unmanaged code. That is, it is not code that runs in the .NET Framework. When IIS routes the request to the aspnet\_isapi.dll ISAPI extension, the ISAPI extension routes the request onto the ASP.NET engine, which is written in managed code - managed code is code that runs in the .NET Framework.

The ASP.NET engine is strikingly similar to IIS in many ways. Just like IIS has a directory mapping file extensions to ISAPI extensions, the ASP.NET engine maps file extensions to HTTP handlers. An HTTP handler is a piece of managed code that is responsible for generating the markup for a particular file type.

**4.5. Integrating the Website and Database**

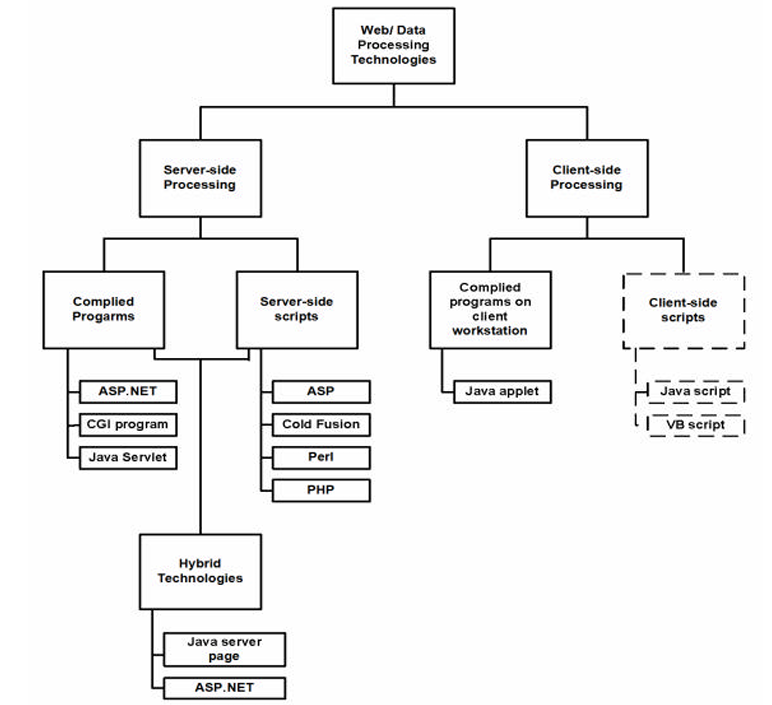
Customers ordering from an e-commerce website need to be able to get information about a vendor’s products and services, ask questions, select items they wish to purchase, and submit payment information. Vendors need to be able to track customer inquiries and preferences and process their orders. So a well organized database is essential for the development and maintenance of an e-commerce site [3]. In a static Web page, content is determined at the time when the page is created. As users access a static page, the page always displays the same information.

Example of a static Web page is the page displaying company information. In a dynamic Web page, content varies based on user input and data received from external sources. We use the term “data-based Web pages” to refer to dynamic Web pages deriving some or all of their content from data files or databases. A data-based Web page is requested when a user clicks a hyperlink or the submit button on a Web page form. If the request comes from clicking a hyperlink, the link specifies either a Web server program or a Web page that calls a Web server program. In some cases, the program performs a static query, such as “Display all items from the Inventory”. Although this query requires no user input, the results vary depending on when the query is made. If the request is generated when the user clicks a form’s submit button, instead of a hyperlink, the Web server program typically uses the form inputs to create a query. For example, the user might select five books to be purchased and then submit the input to the Web server program.

The Web server program then services the order, generating a dynamic Web page response to confirm the transaction. In either case, the Web server is responsible for formatting the query results by adding HTML tags. The Web server program then sends the program’s output back to the client’s browser as a Web page.

**5. Web Page Programming Options**

An e-commerce organization can create data-based Web pages by using server side and client-side processing technologies or a hybrid of the two. With server-side processing, the Web server receives the dynamic Web page request, performs all processing necessary to create the page, and then sends it to the client for display in the client’s browser. Client-side processing is done on the client workstation by having the client browser execute a program that interacts directly with the database.

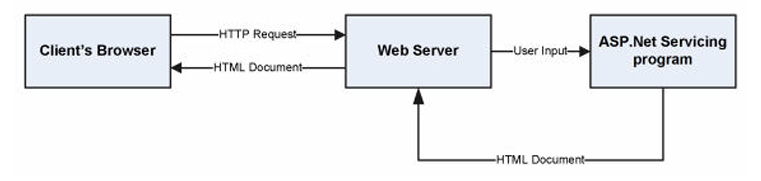


**FIGURE:21 Web page programming options**

outlines commonly used server-side, client-side, and hybrid Web and data processing technologies; client-side scripts are in dashed lines to indicate they are unable to interact directly with a database or file but are used to validate user input on the client, then send the validated inputs to the server for further processing.

**5.1 SERVERSIDE PROCESSING**

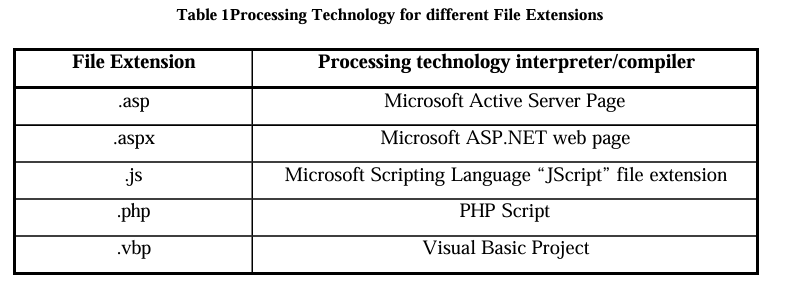
Generally dynamic or data-driven Web pages use HTML forms to collect user inputs, submitting them to a Web server. A program running on the server processes the form inputs, dynamically composing a Web page reply. This program, which is called, servicing program, can be either a compiled executable program or a script interpreted into machine language each time it is run.

****

**FIGURE:22 COMPILED SERVER PROGRAMS FLOW CHART**

ASP.NET has introduced many new capabilities to server-side Web programming, including a new category of elements called server controls that generate as many as 200 HTML tags and one or more JavaScript [9] functions from a single server control tag. Server controls support the processing of user events, such as clicking a mouse or entering text at either the client browser or the Web server. Server controls also encourage the separation of programming code into different files and/or areas from the HTML tags and text of a Web page, thus allowing HTML designers and programmers to work together more effectively. Server-side scripts. Web-based applications can also use server-side scripts to create dynamic Web pages that are able to retrieve and display information from a backend database and modify data records

If needed, a developer can have a single Web server process a variety of scripts written with any or all of these technologies. The Web server knows which script interpreter to invoke by taking note of the requesting script’s file extension. Table 1 below demonstrates some commonly used extensions and the related technologies



Both ASP.NET and JSP are considered as Hybrid server side technologies. ASP.NET is designed to work under the Windows/Server and IIS web server environment. JSP is more portable as it works in most Operating Environments including Windows and Linux.

**5.2. Client-Side Processing.**

Client-side Web page processing is achievable through compiled programs downloaded, installed, and executed on the client workstation or by creating scripts with the HTML Web page commands interpreted by the client browser. Downloading and running compiled programs on client workstations.

When a user clicks a hyperlink on a Web page associated with a compiled client-side program, the user’s browser must have the ability to run the executable program file; this program interacts with the user, sending and retrieving data from a database server as needed. Many times, the user is asked to install certain ActiveX components to view some animations or play games.

This new component plugs in into the existing system, thus extending the functionality of the system. Java Applets are another example of compiled programs on client workstations. An applet is a program written in the Java programming language that can be included in an HTML page, much in the same way an image is included in a page. When we use a Java technology-enabled browser to view a page that contains an applet, the applet's code is transferred to our system and executed by the browser Client-side scripts. In a client-side script, source code written in such languages as JavaScript and VBScript is embedded in an HTML document. JavaScript is the most commonly used client-side scripting language and is supported by most browsers.

**6. Web Based Application Development**

The Web is built on the HyperText Transfer Protocol. HTTP is a client/server request/reply protocol that is stateless. That is, the protocol does not make any association between one transaction and another; e.g.: time since the last transaction, type or client involved in the last transaction, what data was exchanged between the client and the server. As far as HTTP is concerned, each transaction is a discrete event. But this is not what we want in a shopping cart application because we need to preserve the user’s shopping selection as they proceed with their purchase, in addition it is useful to have the access to their past purchase history and personal preferences.

The Session object is used to store information about, or change settings for a user session. Variables stored in the Session object hold information about one single user, and are available to all pages in one application. Common information stored in session variables are name, id, and preferences. The server creates a new Session object for each new user, and destroys the Session object when the session expires. In this project, the concept of session variables will be used for maintaining state information.

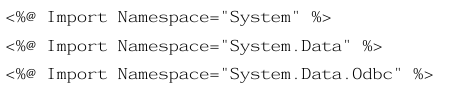
**7. Database Connectivity**

In e-commerce applications it is very typical for the Web server to contact the database to get information as needed. ASP.NET uses a technology called ActiveX Data Objects.NET (ADO.NET) to connect to the database.

**7.1 ADO.NET**

Classic ASP pages used ActiveX Data Objects (ADO) to access and modify databases. ADO is a programming interface used to access data. This method was efficient and fairly easy for developers to learn and implement. However, ADO suffered from a dated model for data access with many limitations, such as the inability to transmit data so it is easily and universally accessible. Coupled with the move from standard SQL databases to more distributed types of data (such as XML), Microsoft introduced ADO.NET. Although ADO.NET is known as the next evolution of ADO, it is very different from its predecessor. Whereas ADO was connection-based, ADO.NET relies on short, XML message-based interactions with data sources. This makes ADO.NET much more efficient for Internet-based applications.

In ADO.NET there are two core objects that allow us to work with data initially: the DataReader and the DataSet. In any .NET data access page, before we connect to a database, we first have to import all the necessary namespaces that will allow us to work with the objects required. Namespace in .NET is a set of classes that can be used while creating an application. The .NET Framework has about 3,500 classes which can be accessed through a namespace. The application will be using a technology known as Open DataBase Connectivity (ODBC) to access the database; therefore we must first import necessary namespaces. Below is a sample namespace declaration used by.NET.



After all the necessary namespaces are imported, a connection to the database is made.

**OdbcConnetion odbcCon = new OdbcConnection ("DRIVER = {MySQL ODBC 3.51 Driver}; SERVER=localhost; DATABASE=project; UID=root; PASSWORD=pwd");**

**odbcCon.Open();**

The above statement creates a connection to the database with an OdbcConnection object. This object tells ASP.NET where to go to get the data it needs. Since the data is stored in the same computer as the application, the SERVER is given as localhost. Next we open the connection object. Listed below are the common connection object methods we could work with:

**• Open** - Opens the connection to our database

**• Close** - Closes the database connection

**• Dispose** - Releases the resources on the connection object. Used to force garbage collecting, ensuring no resources are being held after our connection is used.

**• State** - Tells you what type of connection state your object is in, often used to check whether the connection is still using any resources.

Once the connection is made, in order to access the data in a database, ADO.NET relies on two components: DataSet and Data Provider [20]. These components are explained below.

**DataSet**

The dataset is a disconnected, in-memory representation of data. It can be considered as a local copy of the relevant portions of the database. The DataSet resides in memory and the data in it can be manipulated and updated independent of the database. If necessary, changes made to the dataset can be applied to the central database. The data in DataSet can be loaded from any valid data source such as a text file, an XML database, Microsoft SQL server database, an Oracle database or MySQL database.

**Data Provider**

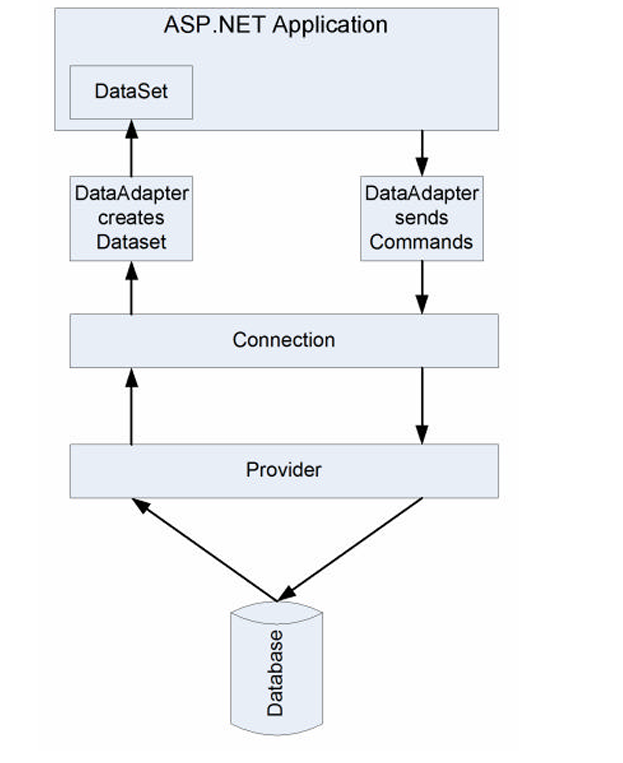
The Data Provider is responsible for providing and maintaining the connection to the database. A DataProvider is a set of related components that work together to provide data in an efficient and performance driven manner. Each DataProvider consists of the following component classes:

• The Connection object which provides a connection to the database

• The Command object which is used to execute a command

• The DataReader object which provides a read only, connected recordset

• The DataAdapter object which populates a disconnected DataSet with data and performs the update.



**FIGEURE:23 ADO.NET ARCHITECTURE**

• The Connection object establishes a connection to the database and submits the request sent by DataAdapter.

• The Connection object connects to the database through a Provider such as ODBC.NET. The Provider acts as a translator between the Connection object and the database. It translates the request for data to database’s language and brings back the data, if needed.

• The Provider sends the data back to the DataAdapter through the Connection object and DataAdapter places the data in a DataSet object residing in application’s memory.

**7.2 Connecting ASP.NET**

application to a Database The steps required to connect our ASP.NET application to the MySQL database and access the data are given below:

**1.Import the required namespaces**.

using System;

using System.Data;

using SystemData.Odbc;

**2.Create a connection object**.

string myConnectionString;

myConnectionString = “DRIVER = {MySQL ODBC 3.51 Driver}; SERVER = localhost; DATABASE = project; UID = root; PASSWORD = ‘’” OdbcConnection odbcCon = new odbcConnection(myConnectionString)

**3.create a sql query**

string str; str="Select \* from Customer where UserID='admin’;

**4.Create a Command object to run the SQL query**

odbcCmd=new OdbcCommand(strodbcCon);

**5. DataReader to read the result**

OdbcDataReader odbcReader; String text, text2;

while (odbcReader.Read())

{

text = odbcReader["UserID"]. ToString ();

text2 = odbcReader[“FirstName”] To String ();

}

**6.Close odbcReader and odbcConnection**

odbcReader.Close(); odbcCon.Close();

The data can now be used as desired by the application.

**8. The Shopping Cart Application**

The objective of this application is to provide the user an online website where they can buy books from the comfort of their home. A shopping cart is used for the purpose. The user can select the desired books, place them in the shopping cart and purchase them using a Credit Card. The user’s order will be shipped according to the type of shipping selected at the time of placing the order. Website consists of the following web pages:

1. AddBook.aspx

2. BookDetails.aspx

3. BookReview.aspx

4. Books.aspx

5. ChangePassword.aspx

6. CheckOut.aspx

7. FinalOrder.aspx

8. Footer.ascx

9. ForgotPassword.aspx

10. Login.aspx

11. LogOff.aspx

12. Menu.ascx

13. Order.aspx

14. PurchaseHistory.aspx

15. Registration.aspx

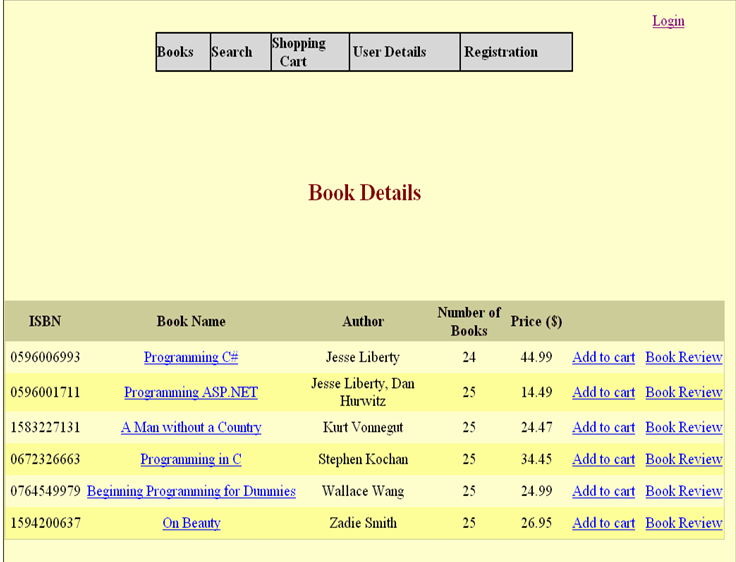
16. Search.aspx

17. ShoppingCart.aspx

18. UserDetails.aspx

Below figures show some screenshots taken from running the application. All the functionalities are explained accordingly.

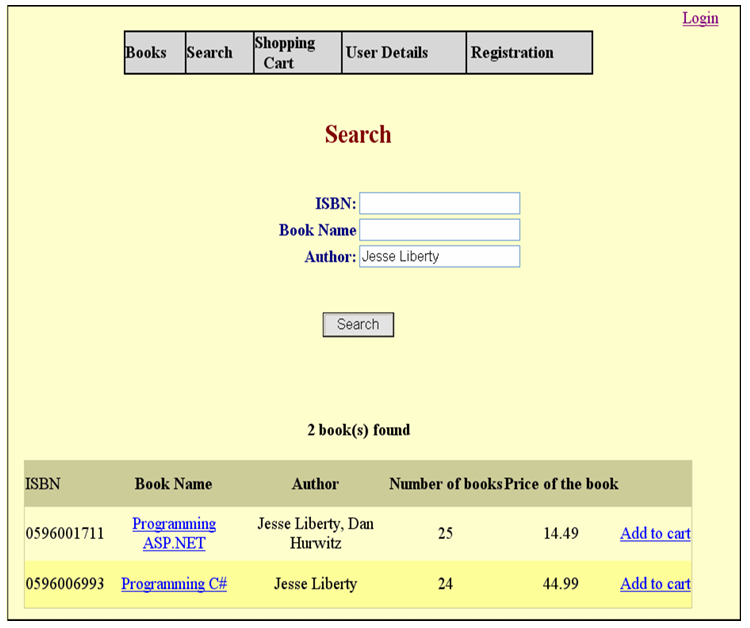
When the user types the web address in the browser, the main page of the application is displayed which has the list of the top ten popular books available in the store, as shown in Figure 24.



**FIGURE:24 BOOK DETAILES**

**8.1. Search for Books**

Books can be searched based on the ISBN, Title or the Author of the book. When searching for books by author “Jesse Liberty”, two books are displayed as shown in figure 24.



**FIGURE:25 SEARCH FOR BOOKS**

**8.2. Registration**

A new user can register on the site by clicking on the registration button on the menu at the top of the page, as shown in Figure 25.

****

**FIGURE 26: NEW USER REGISTRATION**

The “\*” beside the label indicates the required field for successful registration on the site. If the value if not entered, an appropriate message is displayed. If a user with same UserID already exists, the message is displayed. Clicking on Reset will clear all the fields and Submit will submit the information for registration. Upon successful completion, the user is directed to the Books page.

**8.3. User Details**

On clicking “User Details”, the detailed profile information of the user who is currently logged in are displayed as shown in Figure 26.



**FIGURE:27 USER DETAILES**

**8.4. Shopping Cart**

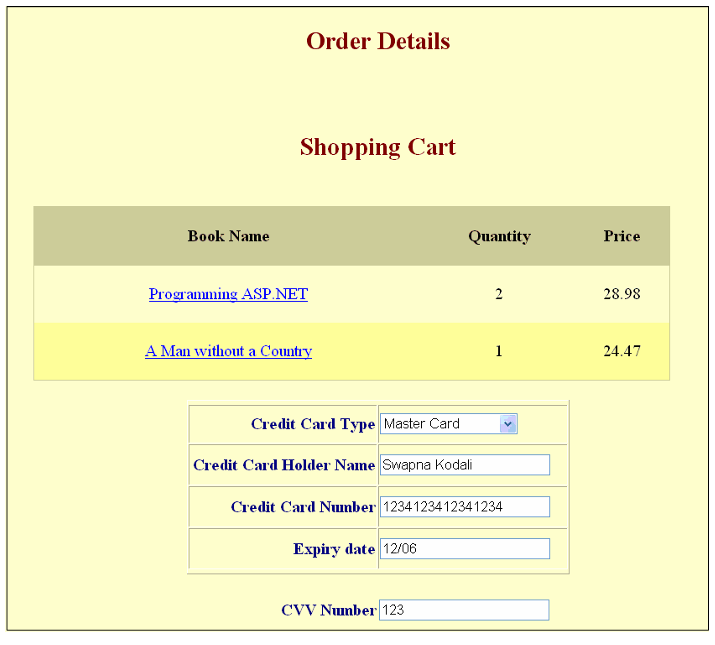
When “Add to Cart” is clicked for any book, it is added to the shopping cart illustrated in Figure 29. If that particular book is already present in the shopping cart, the quantity is increased by 1 and the price is changed accordingly; if not, a new entry is made into the table. All the information in the shopping cart is stored in “shopping\_cart\_items” table. Adding a book into the shopping cart does not decrease the quantity of books in the Books table. It is decreased only after an order is placed for the book. So, placing the book in the shopping cart does not guarantee the availability of the book at the time of placing the order.



**FIGURE 28-SHOPPING CART**

**8.5. Place an Order**

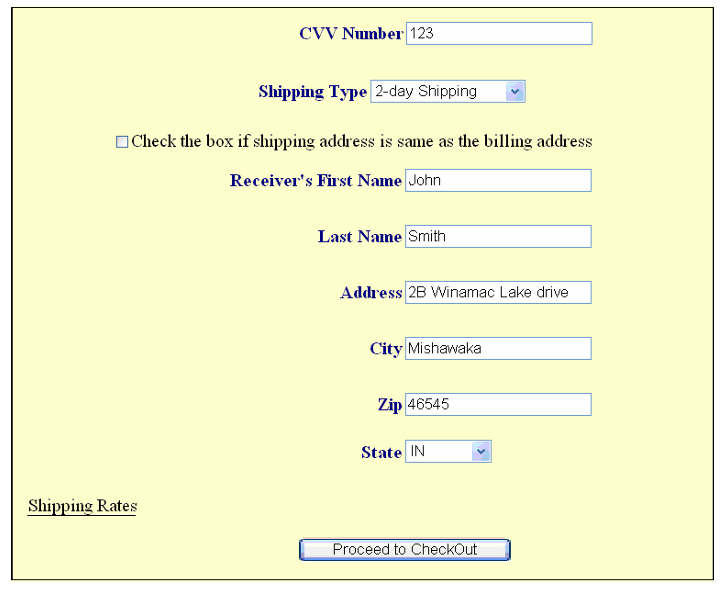
When “Place an Order” button is clicked which is located on the bottom of the shopping cart, the application will ask the user to login if he has not already done so.

****

**FIGURE:29 ORDER DETAILS**

If the user is placing an order with the web site for the first time, they will be asked to enter the credit card details as shown in the above figure; if not, only the Card Verification Value (CVV) number of the credit card is asked for verification, as shown in Figure 29.

At this point, the user can check the shipping address box if shipping address is same as billing address, otherwise the user has to enter the new shipping address as shown in Figure 30.

****

**FIGURE:30-SHIPPING DETAILS**

If the check box provided is checked, the shipping address is obtained from the Customer table. The user also has to select the desired type of shipping for the order. When all the information is entered, the user can “Proceed to the Checkout”.

**8.6 CHECKOUT**

Before placing the final order, the user is shown the total price of the order, which includes total price of books selected, shipping rate and state tax as illustrated in Figure 32. If the user is not satisfied with the order, the order can be cancelled at that point. The information in the shopping cart remains intact, so the user can go back to it and make any changes if necessary. When the “Place Order” button is clicked, the order is placed and the following screen appears which informs the user about the approximate number of days in which the order will be delivered.

****

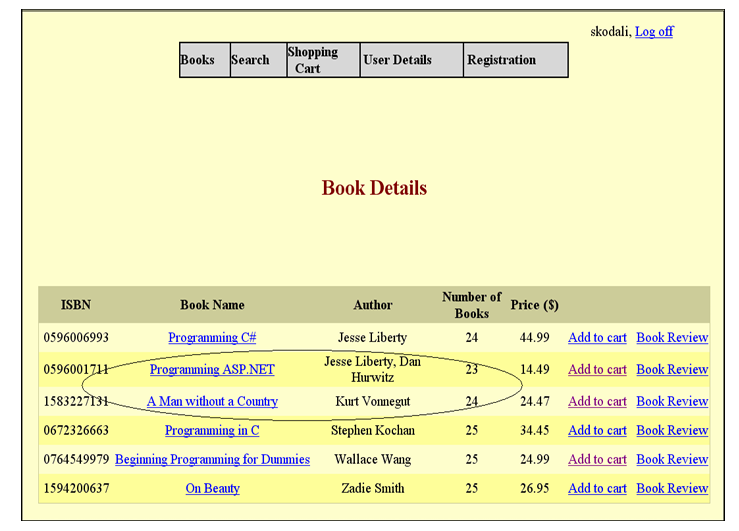
**FIGURE: 31 CHECK OUT**

Once the order is placed, the quantity of the books is reduced in the Books table. The shopping cart for the user cleared and an appropriate message is displayed, as shown in Figure 32.

****

**FIGURE:32 ORDER CONFIRMATION**

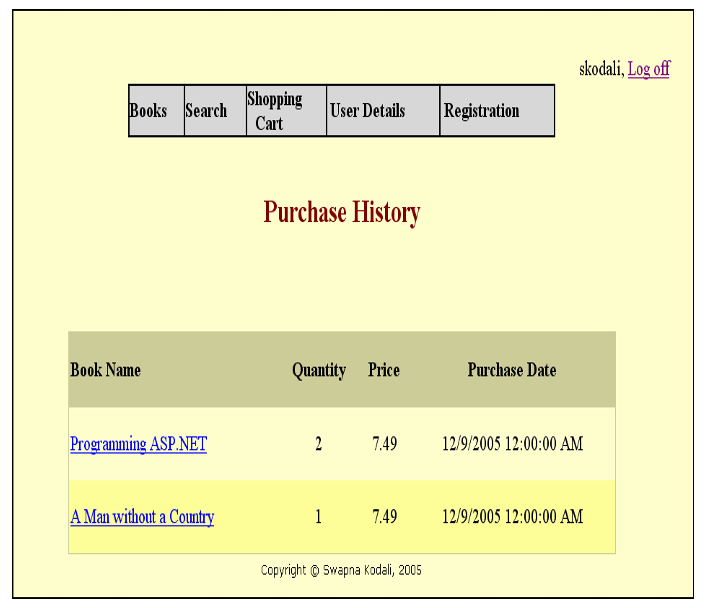
The inventory is updated as shown in Figure 33 after the order is placed.

****

**FIGURE:33 UPDATED INVENTORY AFTER PLACEMENT**

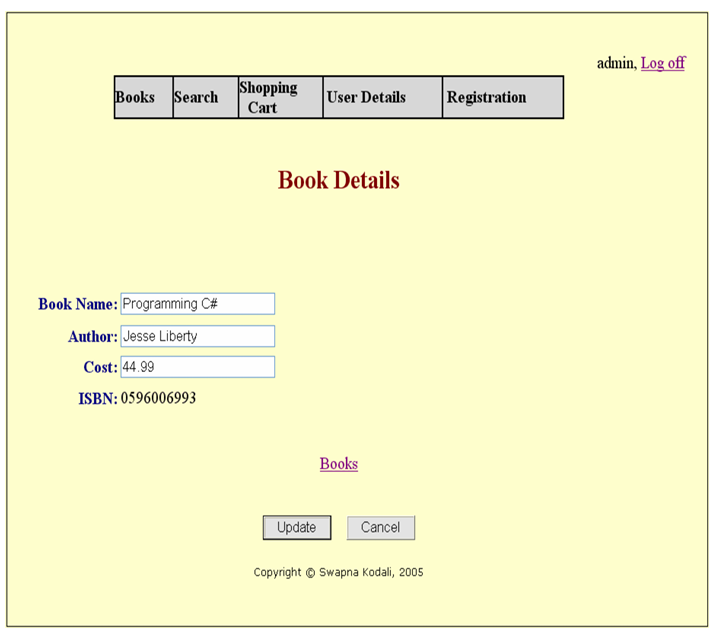
**8.7. Purchase History**

Figure 35 details the purchase history of the user “skodali”. Purchase history can be reached by clicking on the “Purchase History” tab on “User Details” screen as shown in the Figure 28.

****

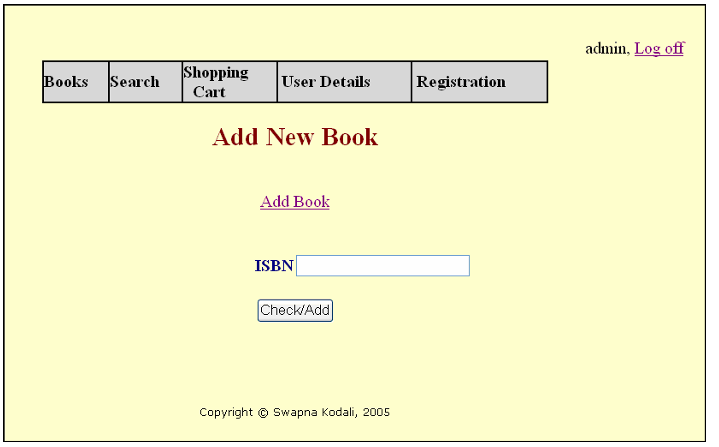
**FIGURE:34 PURCHASE HISTORY**

When viewing the purchase history, the user can view the details of each book by clicking on the book name. The details are displayed as shown in Figure 35.

****

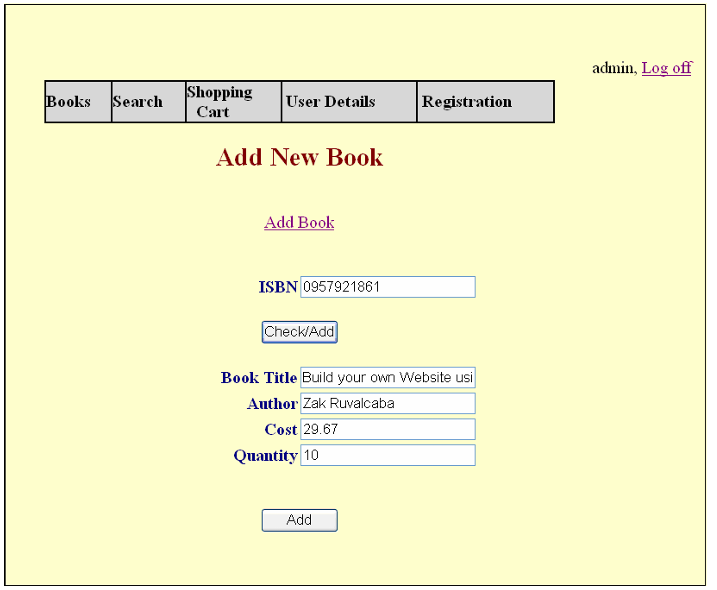
**FIGURE:35 BOOK DETAILS**

Book information can only be changed by the Administrator of the site. All other users can only view the details of the books. The administrator of the site can also “Add Book” or “Remove Book” to/from the Books table. Figure 36 allows a book modification form accessible to the administrator.

****

**FIGURE:36 ADMINISTRATOR MODIFY BOOKS**

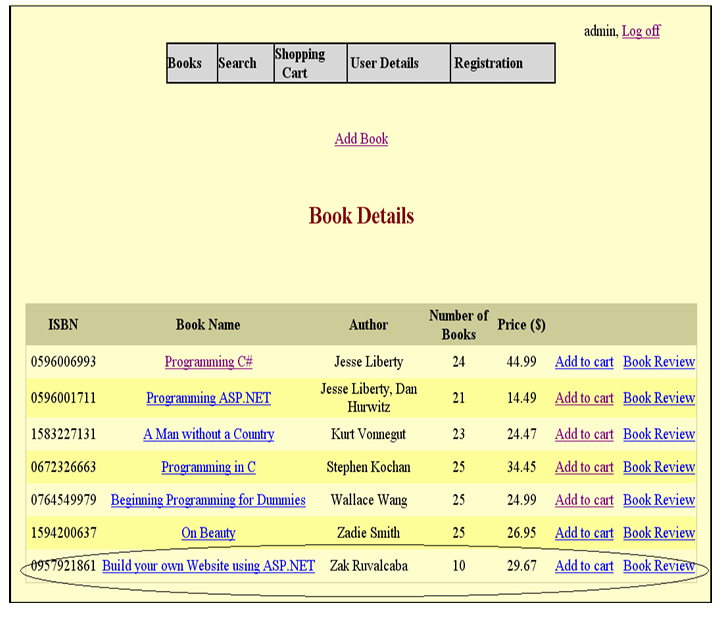
In order to add a book, the administrator will enter the ISBN of the book. If the ISBN is already present in the Books table, the administrator is asked to enter the quantity of the books.

****

**FIGURE:37 DETAILES ABOUT NEW BOOK**

If the entered ISBN book is not present in the Inventory, the administrator is asked to enter the details about the book as shown in Figure 38 to add the book to the inventory.

Figure 38 shows the updated inventory after the book details in Figure 38 are entered. Figure 37 Updated Inventory.

****

**FIGURE:38 UPDATED INVENTORY**

When a user logs off the website, the items in their Shopping Cart are cleared. When the user enters a password during the registration, it is encrypted before it is stored in the database. It is a one-way encryption and the original form cannot be retrieved again. Similar encryption method is used for Credit Card Number. Suppose there are only three “Programming C#” books available in the store. One user adds all of them to the shopping cart, and by the time he chooses other books and places the order, another user has already placed an order for two of those books. In that case, the first user comes to know about this at the time of placing the order and he is directed to the shopping cart to make the appropriate changes. As explained earlier, the user need not be logged in to add books to the shopping cart. When the user adds books without logging in, a GUID (Globally Unique Identifier) is obtained from the system and stored in Session["loginid”] variable. This GUID is stored in the shopping\_cart\_items table along with the selected books by the user. If the user logs in, the GUID in the table is replaced with the actual UserID of the user. Session variables are used to transfer data from one page to another. As soon as the user closes the window, the session variables are cleared.

**8.8. Transactions**

in the Application A transaction is a group of database commands that are treated as a single unit. Transaction must pass what is known as the ACID test:

**Atomic:** All operations in the transaction are executed properly or none. In other words, they make up a single unit of work. For example, if a customer moves and a transaction is used to reflect that change in the database, all parts of the address (street, city, state, etc) must be changed as an atomic action, rather than changing street, then city, then state, and so on.

**Consistent:** The execution of a single transaction preserves the consistency of the database. All the relationships between data in a database are maintained correctly. For example, if customer information uses a tax rate from a state tax table, the state entered for the customer must exist in the state tax table.

**Isolation:** Each transaction is unaware of the other transactions occurring concurrently. Changes made by other clients cannot affect the current changes. For example, if two data entry operators try to make a change to the same customer at the same time, one of two things occurs: either one operator's changes are accepted and the other is notified that the changes were not made, or both operators are notified that their changes were not made. In either case, the customer data is not left in an indeterminate state.

**Durability:** Changes the transaction has performed persist in the database. Once a change is made, it is permanent. If a system error or power failure occurs before a set of commands is complete, those commands are undone and the data is restored to its original state once the system begins running again.

Transaction processing is particularly important for Web applications that use data access, since Web applications are distributed among many different clients. In a Web application, databases are a shared resource, and having many different clients distributed over a wide area can present these key problems:

• Contention for resources. Several clients might try to change the same record at the same time. This problem gets worse the more clients you have.

• Unexpected failures. The Internet is not the most reliable network, even if your Web application and Web server are 100 percent reliable. Clients can be unexpectedly disconnected by their service providers, by their modems, or by power failures.

• Web application life cycle. Web applications do not follow the same life cycle as Windows applications—Web forms live for only an instant, and a client can leave your application at any point by simply typing a new address in their browser.

Transaction processing follows these steps:

1. Begin a transaction.

2. Process database commands.

3. Check for errors.

4. If errors occurred, restore the database to its state at the beginning of the transaction. If no errors occurred, commit the transaction to the database.

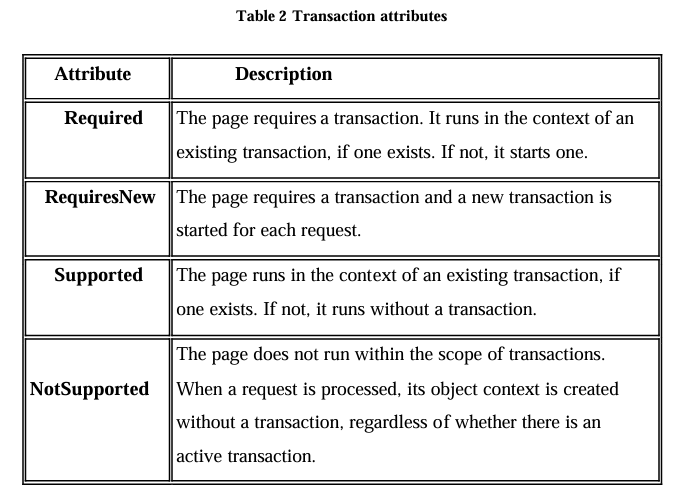
Suppose two users try to add the same book to the shopping cart and try to place an order at the exact same time. An update should be done to the Books table after the order is placed, but if only the latest transaction is noted down, the book quantity will differ in the real world.

This situation has to be handled as in a “Transaction”. As detailed earlier, a transaction is an operation or set of operations that succeeds or fails as a logical unit. That is, either both the updates are not done, or both the updates are done consecutively. Transactions are normally managed by declaring boundaries around a set of operations. Operations that execute in the context of the transaction boundary then succeed or fail as a unit.

For ASP.NET, the transaction boundary is the execution of a single request to a page, which might contain nested components that participate in the same transaction. While the page is executing, if an operation on the page itself or a nested component in the same transaction fails, it can call ContextUtil.SetAbort. This is then picked up by the current transaction context, the entire transaction fails, and any operations that were already completed are undone. If nothing fails, the transaction is committed. ASP.NET support for transactions consists of the ability to allow pages to participate in ongoing Microsoft .NET Framework transactions. Transaction support is exposed via an @Transaction directive that indicates the desired level of support:



Table 2 defines the supported transaction attributes. The absence of a transaction directive is the same as an explicit directive to "Disabled”



A transaction can be explicitly committed or aborted using static methods of the System. Enterprise Services.ContextUtil class. You can explicitly call the SetComplete or SetAbort method to commit or abort an ongoing transaction.

A transaction will commit or abort at the end of the page's lifetime depending on whether SetComplete or SetAbort was called last, provided there is no other object to join the same transaction.

**9.Limitations and Future Development**

There are some limitations for the current system to which solutions can be provided as a future development:

1. The system is not configured for multi-users at this time. The concept of transaction can be used to achieve this.

2. The Website is not accessible to everyone. It can be deployed on a web server so that everybody who is connected to the Internet can use it.

3. Credit Card validation is not done.

Third party proprietary software can be used for validation check. As for other future developments, the following can be done:

1. The Administrator of the web site can be given more functionalities, like looking at a specific customer’s profile, the books that have to be reordered, etc.

2. Multiple Shopping carts can be allowed.

**10. Conclusion**

The Internet has become a major resource in modern business, thus electronic shopping has gained significance not only from the entrepreneur’s but also from the customer’s point of view. For the entrepreneur, electronic shopping generates new business opportunities and for the customer, it makes comparative shopping possible. As per a survey, most consumers of online stores are impulsive and usually make a decision to stay on a site within the first few seconds. “Website design is like a shop interior. If the shop looks poor or like hundreds of other shops the customer is most likely to skip to the other site”[16]. Hence we have designed the project to provide the user with easy navigation, retrieval of data and necessary feedback as much as possible.

In this project, the user is provided with an e-commerce web site that can be used to buy books online. To implement this as a web application we used ASP.NET as the Technology. ASP.NET has several advantages such as enhanced performance, scalability, built-in security and simplicity. To build any web application using ASP.NET we need a programming language such as C#, VB.NET, J# and so on. C# was the language used to build this application. For the client browser to connect to the ASP.NET engine we used Microsoft’s Internet Information Services (IIS) as the Web Server. ASP.NET uses ADO.NET to interact with the database as it provides in-memory caching that eliminates the need to contact the database server frequently and it can easily deploy and maintain an ASP.NET application.

MySQL was used as back-end database since it is one of the most popular open source databases, and it provides fast data access, easy installation and simplicity. A good shopping cart design must be accompanied with user-friendly shopping cart application logic. It should be convenient for the customer to view the contents of their cart and to be able to remove or add items to their cart. The shopping cart application described in this project provides a number of features that are designed to make the customer more comfortable.

This project helps in understanding the creation of an interactive web page and the technologies used to implement it. The design of the project which includes Data Model and Process Model illustrates how the database is built with different tables, how the data is accessed and processed from the tables. The building of the project has given me a precise knowledge about how ASP.NET is used to develop a website, how it connects to the database to access the data and how the data and web pages are modified to provide the user with a shopping cart application.

**11.Bibliography**

**Articles**

1. Chen, L. (2000). Enticing Online Consumers: A Technology Acceptance Perspective Research-in-Progress. ACM Proceedings, SIGCPR.

2. Diwakar, H., Marathe, M. (2000). The architecture of a one-stop web-window shop. December, ACM SIGecom Exchanges, Volume 2 Issue 1.

3. Morrison, M., Morrison, J., and Keys, A. (2002). Integrating Web Sites and Databases. Communications of the ACM, September, Volume 45, Issue 9.

4. Kubilus, N. J. (2000). Designing an e-commerce site for users. September 2000, Crossroads, Volume 7 Issue 1.

5. Norman, D.A. The Design of Everyday Things. Doubleday, New York, 1994.

6. Tilson, R., Dong, J., Martin, S., Kieke, E. (1998). A comparison of two current e commerce sites. September, Proceedings of the 16th annual international conference on Computer documentation.

**Books**

7. Anderson, R., Francis, B., Homer, A., Howard, R., Sussman, D. and Watson. (2001) Professional ASP.NET. Wrox Press Ltd.

8. Brown, S., Burdick, R., Falkner, J., Galbraith, B., Johnson, R., Kim, L., Kochmer, C., Kristmundsson, T. and Li S (2001). Professional JSP. Wrox Press Ltd.

9. Walther, S. (1998) Active Server Pages. SAMS Net.

10. Wagner, R., Daniels, K., Griffin, G., Haddad, C. and Nasr, J. (1997) JavaScript Unleashed. SAMS Net.

11. Wiley, Y. M. J. & Sons. (1997) Creating the Virtual Store: Taking Your Web Site from Browsing to Buying.

**12 .Websites**

12. http://encyclopedia.laborlawtalk.com/IIS for information on II

13. http://aspnet.4guysfromrolla.com/articles/020404-1.aspx for relationship between IIS and ASP.NET.

14.http://216.15.201.66/dpec/course.htm?fullpg=http%3A//216.15.201.66/dpec/cours es/wac312/wah006.htm&acro=wac312 for security authentication in ASP.NET

15.http://samples.gotdotnet.com/quickstart/aspplus/doc/mtstransactions.aspx for information on Transactions in ASP.NET.

16. http://www.x-cart.com/articles/design\_development.html behavior.

17. http://aspnet.4guysfromrolla.com/articles/011404-1.aspx for online customer for relation between IIS and ASP.NET.

18.http://www.informatik.uni-bremen.de/uniform/gdpa\_d/methods/m-fctd.htm definition of Functional Decomposition.

19. http://www.agilemodeling.com/artifacts/dataFlowDiagram.htm for for definition of Data Flow Diagram. 20. http://www.startvbdotnet.com/ado/default.aspx for information on ADO.NET

21. http://mypage.iusb.edu/~hhakimza/505/index.html for ADO.NET objects.

22. http://msdn.microsoft.com for ADO.NET objects.