TABLE OF CONTENT

CHAPTER – 1 (OVERVIEW)

1.1 INTRODUCTION

In today's digital age, where technology permeates every aspect of our lives, the realm of literature stands as a timeless beacon of knowledge, imagination, and cultural enrichment. As the world embraces the convenience and accessibility offered by digital commerce, there arises a compelling need for an online platform that caters specifically to the discerning tastes and voracious appetites of book enthusiasts. It is in response to this burgeoning demand that the "QuillQuest – an Online Bookstore Website" project emerges, poised to revolutionize the way readers discover, engage with, and procure literary treasures

The landscape of literary consumption has undergone a profound transformation, propelled by advancements in technology and the ever-expanding reach of the internet. Today, readers seek more than mere books; they crave immersive experiences, personalized recommendations, and seamless access to an expansive universe of literary works. The "QuillQuest" project arises against this backdrop of evolving reader expectations, aiming to redefine the online bookstore paradigm and set new standards of excellence in the digital realm of literature.

Advantages:

Enhanced Efficiency: By this website improves efficiency, such as automating manual tasks or streamlining processes, it will enable users to accomplish tasks more quickly and effectively.

Improved User Experience: The project will enhance the overall user experience, leading to higher satisfaction and engagement.

Increased Productivity: The implementation of specific features or functionalities will facilitate collaboration, communication, and workflow management, thereby boosting productivity across teams and departments.

Scalability and Flexibility: The architecture and design of the software will be scalable and flexible, allowing for how the system can adapt to changing requirements, handle increased loads, or integrate with third-party systems, ensuring long-term viability and growth.

Cost Savings: Through cost-saving measures, such as resource optimization, automation, or open-source technologies, the project will help reduce operational expenses and maximize return on investment (ROI) for stakeholders.

1.2 BACKGROUND

The inception of "QuillQuest" was sparked by a shared passion for literature and a desire to create a digital space where readers could immerse themselves in the boundless world of books. Drawing inspiration from the transformative power of storytelling, I envisioned a platform that would not only facilitate the buying and selling of books but also serve as a hub for literary discovery, community engagement, and intellectual exploration.

Extensive market research revealed a growing demand for online platforms that offer a diverse and personalized selection of books, catering to the evolving tastes and preferences of readers worldwide. With the proliferation of e-commerce and digital publishing, there existed a unique opportunity to leverage technology to enhance the book-buying experience and foster deeper connections between readers and authors.

The motivation behind "QuillQuest" stems from a shared belief in the transformative impact of literature on individuals and society at large. By democratizing access to books and fostering a sense of community among readers, the project aims to promote literacy, ignite imaginations, and celebrate the universal language of storytelling. With a commitment to inclusivity, innovation, and excellence, "QuillQuest" seeks to inspire a lifelong love of reading and learning in people of all ages and backgrounds.

In summary, the background of "QuillQuest" is rooted in a deep appreciation for the written word and a vision to create a digital ecosystem where literature thrives and readers thrive alongside it. With a firm understanding of market trends, consumer needs, and the transformative power of storytelling, "QuillQuest" is poised to revolutionize the online bookstore experience and leave an indelible mark on the world of literature.

1.3 OBJECTIVE

The objectives of the "QuillQuest – an Online Bookstore Website" project are multifaceted, aiming to create an inclusive and enriching digital environment for readers worldwide. Firstly, the project seeks to enhance accessibility by providing a seamless browsing experience across diverse genres, languages, and devices, ensuring that readers can easily discover and engage with a vast array of literary works. Personalization is also a key focus, with the aim of tailoring recommendations based on individual preferences and browsing habits, thus enriching the discovery process and fostering deeper connections between readers and books. Furthermore, the project aspires to foster a vibrant community of readers, authors, and enthusiasts, facilitating dialogue, collaboration, and the sharing of literary experiences through social features and user-generated content. Technological innovation is integral to the project's success, with a commitment to leveraging cutting-edge technologies such as machine learning and artificial intelligence to enhance the user experience and streamline processes. Finally, the project aims for sustainable growth and business success through strategic partnerships, revenue diversification, and customer retention initiatives, ensuring that QuillQuest remains a trusted and indispensable destination for book lovers worldwide.

1.4 PURPOSE

The purpose of the "QuillQuest – an Online Bookstore Website" project is to address the evolving needs and expectations of book enthusiasts in today's digital landscape. With the proliferation of digital commerce and technological advancements, there exists a compelling opportunity to create an innovative and user-centric platform that redefines the online bookstore experience. QuillQuest aims to serve as more than just a virtual marketplace for books; it aspires to be a dynamic and inclusive community where readers can discover, engage with, and celebrate the transformative power of literature.

The purpose of the "QuillQuest – an Online Bookstore Website" project is to create a user-friendly and efficient platform for book enthusiasts to browse, purchase, and engage with literature online. The project aims to provide a convenient and accessible way for users to explore a wide range of books, including fiction, non-fiction, academic, and specialized genres, from the comfort of their own homes.

1.5 SCOPE

The scope of this project is comprehensive, encompassing the creation of a digital platform that redefines the online bookstore experience. The project will involve curating an extensive catalog of literary works spanning various genres, languages, and cultural backgrounds, ensuring a diverse selection to cater to the tastes of readers worldwide. QuillQuest will prioritize an intuitive and user-friendly interface, featuring advanced search functionalities and personalized recommendations powered by machine learning algorithms. Additionally, the project will focus on fostering community engagement through interactive features such as discussion forums, book clubs, and author interviews, creating a dynamic digital ecosystem for readers to connect and share their love for literature. Seamless e-commerce functionality will be integrated to facilitate secure transactions, efficient order management, and hassle-free shopping experiences for customers. Leveraging cutting-edge technologies and scalable architecture, QuillQuest aims to deliver a high-performance platform capable of handling growing user demands and staying at the forefront of the digital publishing landscape. In summary, the scope of the project encompasses content curation, user experience design, community engagement, e-commerce functionality, technological innovation, and scalability, with the overarching goal of empowering readers to explore, discover, and connect with the world of literature in exciting new ways.

LIMITATIONS

While the "QuillQuest – an Online Bookstore Website" project holds promise in transforming the online bookstore experience, it is essential to acknowledge and address potential limitations that may arise during its development and implementation. Firstly, the availability of content poses a challenge, as acquiring rights to certain titles or accessing content from specific publishers or regions may be limited. Additionally, technical constraints such as website performance, scalability, and compatibility across different devices and browsers may present challenges that require careful planning and optimization. Maintaining the privacy and security of user data also remains a priority, yet the risk of data breaches or cyberattacks cannot be entirely eliminated. Resource limitations, including budget, time, and human resources, may impact the project's scope and timeline. Moreover, the highly competitive nature of the online bookstore market, coupled with factors influencing user adoption and engagement, such as changing consumer behaviors and market trends, may pose challenges in gaining visibility and differentiation. Recognizing and addressing these limitations will be crucial to ensuring the success and sustainability of the "QuillQuest – an Online Bookstore Website" project in the dynamic landscape of digital commerce and publishing.

1.6 APPLICABILITY

This project holds significant applicability in addressing the evolving needs and preferences of book enthusiasts in today's digital age. By leveraging innovative technologies and user-centric design principles, QuillQuest aims to offer a transformative online bookstore experience that caters to a diverse audience of readers worldwide.

1. Accessibility and Convenience:
QuillQuest's user-friendly interface and intuitive navigation make it accessible to users of all backgrounds and technological proficiencies. With a comprehensive catalog of literary works spanning genres, languages, and cultural contexts, QuillQuest ensures that readers can easily discover and procure their desired books from the comfort of their homes or on-the-go.

2. Personalized Recommendations:
Through its sophisticated recommendation engine powered by machine learning algorithms, QuillQuest delivers personalized book recommendations tailored to each user's unique preferences and reading habits. By analyzing user behavior and browsing patterns, QuillQuest enhances the discovery process, enabling readers to explore new authors, genres, and literary treasures that resonate with their interests.

3. Community Engagement:
QuillQuest fosters a vibrant and inclusive community of readers, authors, and enthusiasts through its interactive features and social engagement platforms. From discussion forums and book clubs to author interviews and virtual events, QuillQuest provides opportunities for readers to connect, share their love for literature, and engage in meaningful dialogue with like-minded individuals from around the world.

4. Secure and Seamless Transactions:
With robust e-commerce functionality and secure payment gateways, QuillQuest ensures that transactions are conducted safely and efficiently. From browsing and selection to checkout and delivery, QuillQuest prioritizes user trust and satisfaction, providing a seamless shopping experience that instills confidence and peace of mind in its users.

5. Technological Innovation and Scalability:
QuillQuest embraces technological innovation and scalability to deliver a high-performance platform that meets the evolving needs of its users and the demands of the digital marketplace. By leveraging cutting-edge technologies and scalable architecture, QuillQuest remains agile and adaptable, continuously evolving to stay ahead of the curve and deliver unparalleled value to its users.

CHAPTER - 2 (FEASIBILITY STUDY AND STAKEHOLDERS)

2.1 FEASIBILITY STUDY

An important outcome of preliminary investigation is to determine whether the proposed system is feasible or not. feasibility study is an assessment conducted during the initial stages of a project to evaluate its viability, practicality, and potential for success. It involves analyzing various factors, including technical, economic, legal, operational, and scheduling considerations, to determine whether the project is feasible or achievable within the constraints of the available resources and objectives. The primary goal of a feasibility study is to provide stakeholders with valuable insights and information to make informed decisions about whether to proceed with the project, modify its scope or approach, or abandon it altogether.

By conducting a comprehensive feasibility study, project stakeholders can gain a thorough understanding of the project's opportunities, challenges, risks, and constraints, enabling them to make well-informed decisions about its feasibility and potential for success. This information serves as a crucial foundation for the project planning and decision-making process, guiding subsequent actions and ensuring the project's alignment with organizational goals and objectives.

There are Some types of Feasibility Study:

- Technical Feasibility Study
- Economic Feasibility Study
- Financial Feasibility Study
- Operational Feasibility Study

1. Technical Feasibility Study

This type of study assesses whether the proposed project is technically feasible. It examines factors such as technology requirements, infrastructure availability, compatibility with existing systems, and the technical expertise needed to execute the project. In this study, a company's capabilities in terms of technical expertise are the main focus.

When the technical feasibility study report is complete, the following factors are vital –

- A business description to evaluate additional factors that affect the study.
- The specific component/ division of the project/ business under assessment.
- The human resources and cost efficiency factors.
- The possible or alternative solutions to the problems

A technical feasibility study evaluates the practicality and viability of implementing a proposed project from a technical perspective. For this project, a technical feasibility study would assess various technical aspects to determine whether the development and implementation of the online bookstore website are feasible. Here's an outline of what would typically be included in a technical feasibility study for this project:

System Requirements Analysis:

Identify and analyze the technical requirements of the online bookstore website.

Technology Assessment:

Evaluate available technologies and tools that can be used to develop and implement the online bookstore website.

Development Approach:
Determine the most suitable development approach or methodology for building the online bookstore website.
Resource Analysis:
Assess the availability of technical resources.
Risk Assessment:
Identify potential technical risks and challenges that may impact the success of the project.
Conclusion and Recommendations:
Summarize the findings of the technical feasibility study and provide recommendations regarding the feasibility of implementing the online bookstore website from a technical perspective.
Make recommendations for adjustments to the technical plan, resource allocation, or risk management strategies to improve the technical feasibility and likelihood of success of the project.

By conducting a technical feasibility study for the "QuillQuest – an Online Bookstore Website" project, stakeholders can gain valuable insights into the technical requirements, challenges, and risks associated with the project, enabling informed decision-making and strategic planning for its development and implementation.

2. Economic Feasibility Study

An economic feasibility study evaluates the financial viability of the project. It involves estimating the costs associated with the project, including initial investments, operational expenses, and potential revenue streams. The study also analyzes the project's financial returns, payback period, return on investment (ROI), and other financial metrics to determine its economic feasibility.

The "QuillQuest – an Online Bookstore Website" project can help small scale publishers or offline shop owners or writers to keep their books online 24/7 on this website leading to great profits as well as it will help readers and necessary peoples to buy books at affordable level and they even don't have to hustle for having book in hand.

The economic feasibility study conducted for this project has provided valuable insights into the financial viability and potential returns on investment associated with the proposed venture. Through comprehensive analysis and evaluation of various economic factors, including costs and revenue projections following conclusions have been drawn:
Cost Analysis: The initial investment required for developing and launching the QuillQuest platform has been carefully estimated, taking into account factors such as software development, infrastructure setup, marketing expenses, and operational costs. While significant upfront investments are anticipated, the long-term benefits and revenue potential outweigh the initial costs.
Revenue Projections: Based on market research and industry trends, conservative revenue projections have been formulated, considering factors such as the size of the target market, projected user adoption rates, average transaction values, and potential revenue streams (e.g., book sales, premium memberships, advertising). The revenue projections indicate promising financial returns over the projected timeline.

In conclusion, the economic feasibility study confirms that the "QuillQuest – an Online Bookstore Website" project holds strong potential for financial success and viability. The projected revenues outweigh the anticipated costs, and the calculated financial metrics indicate attractive returns on investment. With careful planning, strategic execution, and ongoing monitoring, the project is well-positioned to achieve its financial objectives and establish itself as a leading online destination for book enthusiasts.

3. Financial Feasibility Study

A financial feasibility study assesses the financial viability of a proposed project by analyzing its costs, revenue projections, and financial metrics. A study on whether a project is viable after taking into consideration its total costs and probable revenues. If the revenues cover the costs of the project, then the project, then the project is viable. For this project, a financial feasibility study would examine various financial aspects to determine whether the project is financially feasible and has the potential to generate sufficient returns on investment. Here's an outline of what would typically be included in a financial feasibility study for this project:

Cost Estimation:

Identify and estimate the initial costs associated with developing and launching the online bookstore website. This includes costs such as:

Software development, Website design and hosting, Infrastructure setup, Marketing and promotional expenses.

Revenue Projections:

Forecast the potential revenue streams generated by the online bookstore website.

Financial Metrics Analysis:

Calculate financial metrics to evaluate the project's financial performance and viability.

Sensitivity Analysis:

Conduct sensitivity analysis to assess the project's sensitivity to changes in key assumptions or variables.

Risk Assessment:

Identify and evaluate potential financial risks and uncertainties that may affect the project's financial feasibility. This includes risks related to market conditions, competition, technological changes, regulatory compliance, and financial constraints.

Conclusion and Recommendations:

Summarize the findings of the financial feasibility study and provide recommendations regarding the viability of the project.

Determine whether the project is financially feasible, considering factors such as the projected returns on investment, risk factors, and sensitivity to key variables.

Make recommendations for adjustments to the project plan, financial assumptions, or risk mitigation strategies to improve its financial feasibility and likelihood of success.

By conducting a financial feasibility study for the "QuillQuest – an Online Bookstore Website" project, stakeholders can gain valuable insights into its financial viability, risks, and potential returns, enabling informed decision-making and strategic planning for the project's development and implementation.

4. Operational Feasibility Study

Operational feasibility study is mainly concerned with issues like whether the system will be used if it is developed and implemented. For this project, an operational feasibility study would assess various operational aspects to determine whether the implementation of the online bookstore website is feasible and compatible with the organization's operations. Here's an outline of what would typically be included in an operational feasibility study for this project:

Current Operations Analysis:

Analyze the existing operational processes, workflows, and systems within the organization. This includes: Understanding how the organization currently manages its book inventory, sales, and customer interactions (if applicable).

Identifying any manual or automated processes currently in place for handling book orders, payments, shipping, and customer support.

Impact Assessment:

Assess the potential impact of implementing the online bookstore website on existing operations.

Resource Availability:

Evaluate the availability of resources needed to support the implementation of the online bookstore website.

Training and Support Requirements:

Identify training needs and support requirements for staff who will be involved in managing and operating the online bookstore website. This includes:

Training programs and Technical support.

Risk Assessment:

Identify potential operational risks and challenges that may impact the success of the project.

Conclusion and Recommendations:

Summarize the findings of the operational feasibility study and provide recommendations regarding the feasibility of implementing the online bookstore website from an operational perspective.

Determine whether the project is operationally feasible, considering factors such as impact assessment, resource availability, training and support requirements, and risk mitigation strategies.

Make recommendations for adjustments to operational processes, resource allocation, or support mechanisms to improve the operational feasibility and likelihood of success of the project.

By conducting an operational feasibility study for the "QuillQuest – an Online Bookstore Website" project, stakeholders can gain valuable insights into the operational requirements, challenges, and risks associated with the project, enabling informed decision-making and strategic planning for its implementation.

2.2 STAKEHOLDERS

Stakeholders for this project are individuals, groups, or entities who have an interest or are impacted by the project. Identifying stakeholders is crucial for ensuring their needs and concerns are considered throughout the project lifecycle. Here are some potential stakeholders for the project:

Customers:
Readers and book enthusiasts who will use the online bookstore website to discover, purchase, and engage with books.

Authors:
Writers and authors whose books are featured on the website. They may have an interest in reaching a broader audience and promoting their work.

Management and Investors:
Executives, managers, and investors within the organization funding the project. They have a financial stake in the success of the project and may provide guidance and resources.

Development Team:
Software developers, designers, and technical experts responsible for building and maintaining the online bookstore website.

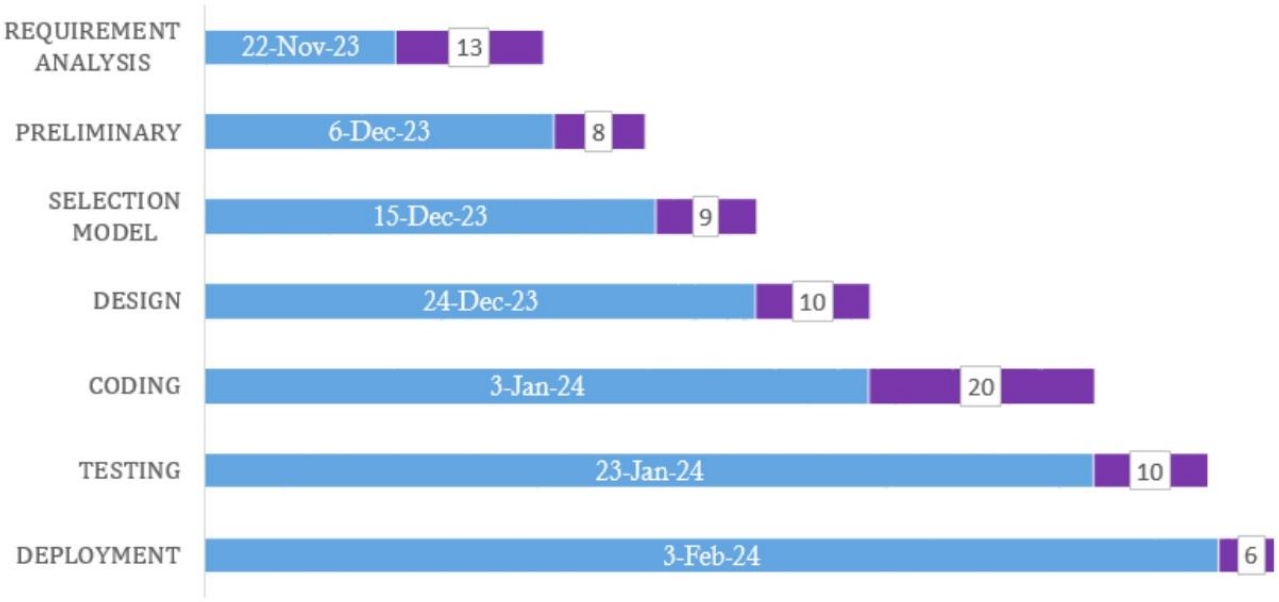
Content Providers:
Publishers, distributors, and content creators who supply books and other literary content for the website.

Payment Processors:
Companies or financial institutions providing payment processing services for online transactions conducted on the website.

Shipping and Logistics Partners:
Shipping companies and logistics providers responsible for delivering orders to customers.

Identifying and engaging with these stakeholders throughout the project lifecycle can help ensure their interests are considered, expectations are managed, and the project's success is maximized.

CHAPTER- 3 (GANTT CHART)



CHAPTER– 4 (SOFTWARE REQUIREMENT SPECIFICATION)

4.1 SOFTWARE AND HARDWARE REQUIREMENTS

Hardware Requirements

Component	Description
Computer/Laptop	Minimum: Dual-core processor, 4GB RAM, 100GB HDD Recommended: Quad-core processor, 8GB RAM, SSD
Monitor	Resolution:1920x1080 or higher
Internet Connection	Broadband or high-speed internet

Software Requirements

Component	Description
Operating System	Windows, MacOS or any type of OS
Java Development Kit	JDK 8 or later
NetBeans IDE	Neatbeans 8.2 or later , here I'm using NetBeans IDE 8.0.2
Application Server	Apache Tomcat 9 or later / Glassfish Server
Database	MySQL 5.7 or later

4.2 FUNCTIONAL REQUIREMENTS

- **User Management**

User Registration: Allow users to register for an account by providing necessary details such as name, email address, and password.
User Authentication: Authenticate users during login using credentials stored securely in the database.
User Profile Management: Enable users to view and update their profile information, including shipping addresses and payment methods.

- **Book Management**

Book Catalog: Display a searchable catalog of available books with details such as title, author, genre, price, and availability.
Book Search: Allow users to search for books based on title, author, genre, or keyword.
Book Details: Provide detailed information about each book, including a synopsis, reviews, and related recommendations.
Shopping Cart: Enable users to add books to a shopping cart, update quantities, and proceed to checkout.

- **Ordering and Checkout**

Order Placement: Allow users to place orders for selected books, specifying shipping address and payment method.
Order Confirmation: Send confirmation emails to users upon successful order placement, including order details and estimated delivery time.
Payment Processing: Integrate with a secure payment gateway to process payments securely using credit/debit cards or other accepted methods.

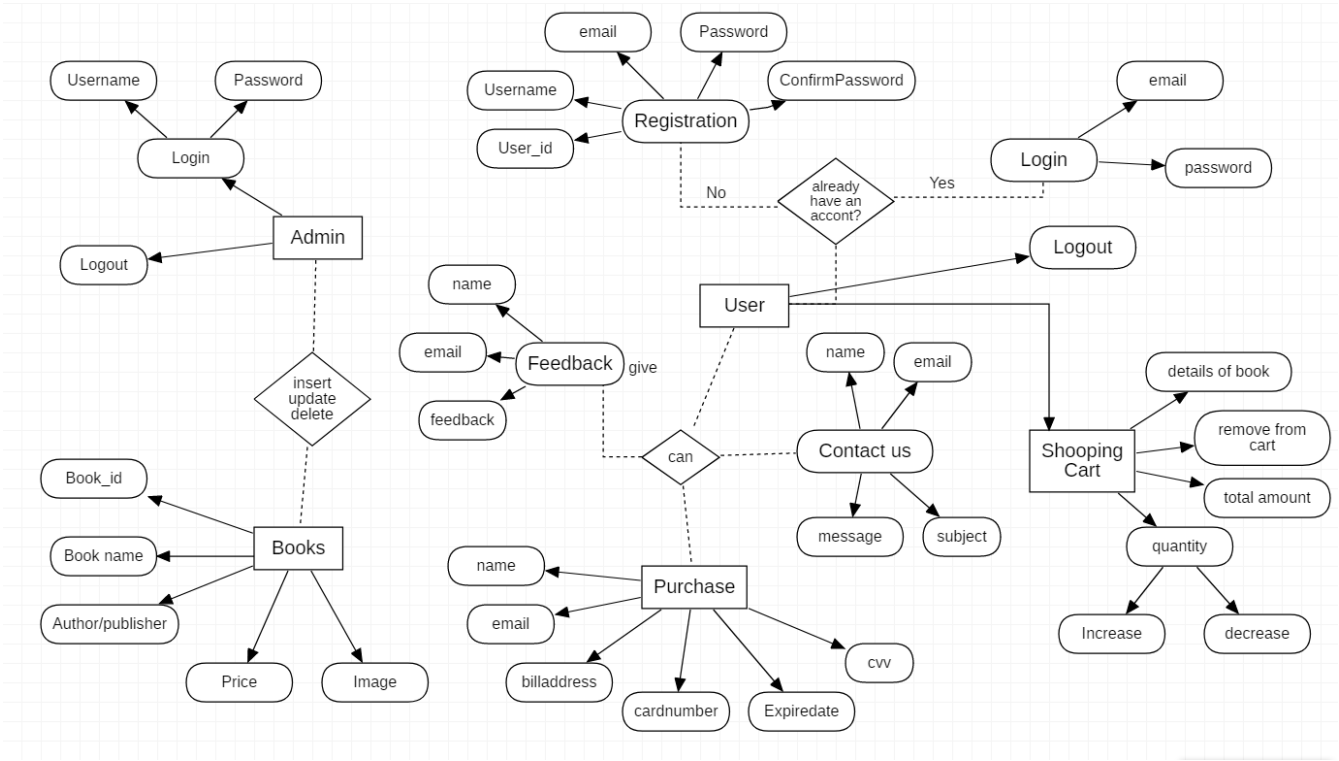
4.3 NON-FUNCTIONAL REQUIREMENTS

Performance: The website shall be responsive and capable of handling concurrent user sessions without significant slowdowns.
Security: Implement appropriate security measures to protect user data, including encryption of sensitive information and protection against common web vulnerabilities.
Scalability: Design the system to scale horizontally to accommodate increasing traffic and growing database size.
Usability: Ensure the website has an intuitive user interface, with clear navigation and user-friendly features to enhance the overall user experience.
Reliability: Minimize system downtime and errors through robust error handling, logging, and monitoring mechanisms.

4.4 CONCEPTUAL MODELS (DATA FLOW DIAGRAMS)

1. E-R Diagram

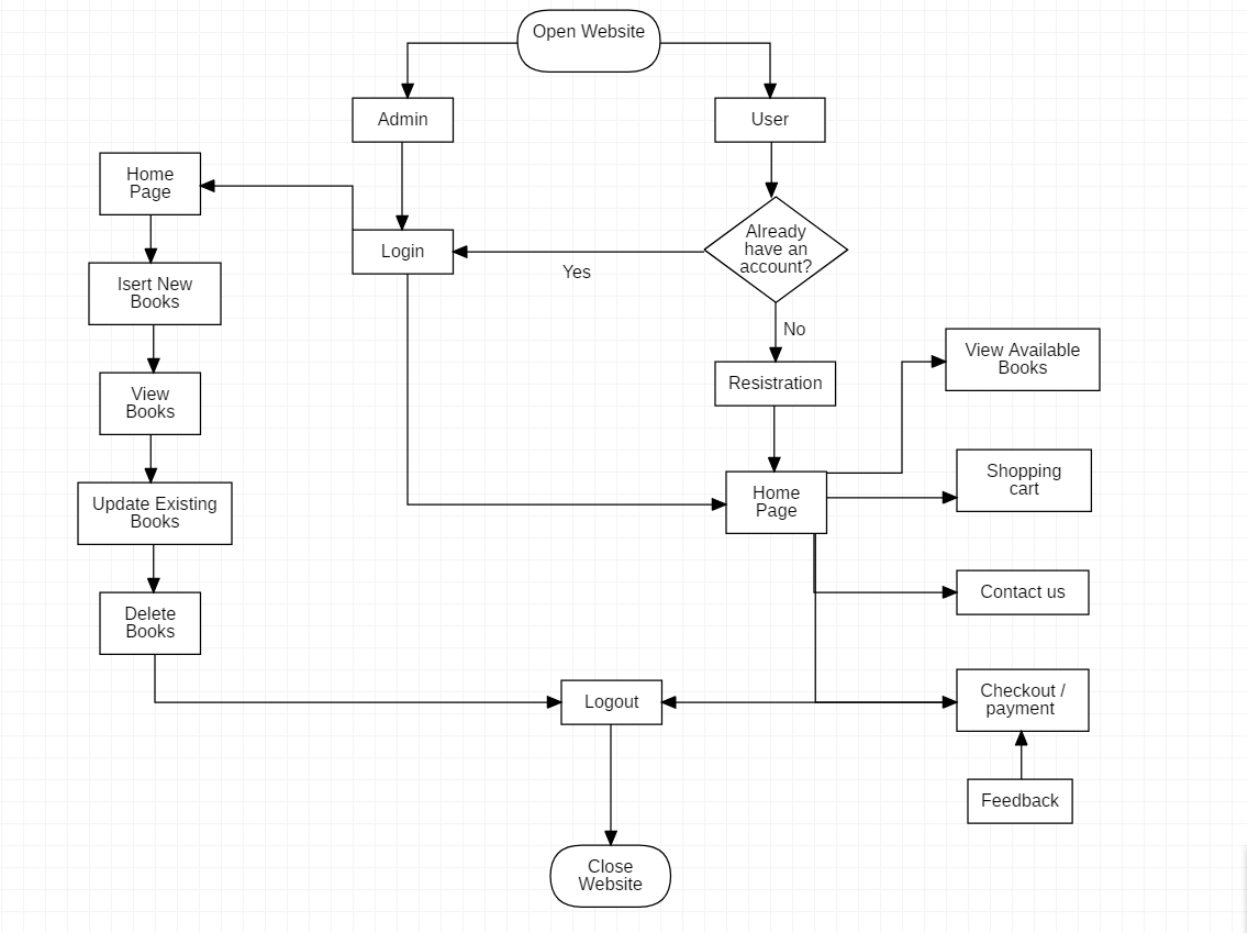
An Entity-Relationship (ER) diagram is a visual representation of the entities, attributes, relationships, and constraints within a database schema. It is a conceptual modeling technique used in database design to depict the structure and organization of data in a system. ER diagrams are essential for designing databases, as they provide a clear and concise overview of the data model and help stakeholders understand the relationships between different entities.



2. Flowchart Diagram

A flowchart diagram is a graphical representation of a process or algorithm, depicting the sequence of steps and decisions involved in completing a task or solving a problem. Flowcharts use standardized symbols and shapes to represent different types of actions, processes, and decision points, making them easy to understand and interpret by various stakeholders.

Overall, flowchart diagrams are versatile tools used in a wide range of fields and industries to visually represent processes, algorithms, and workflows, facilitating communication, analysis, and problem-solving.

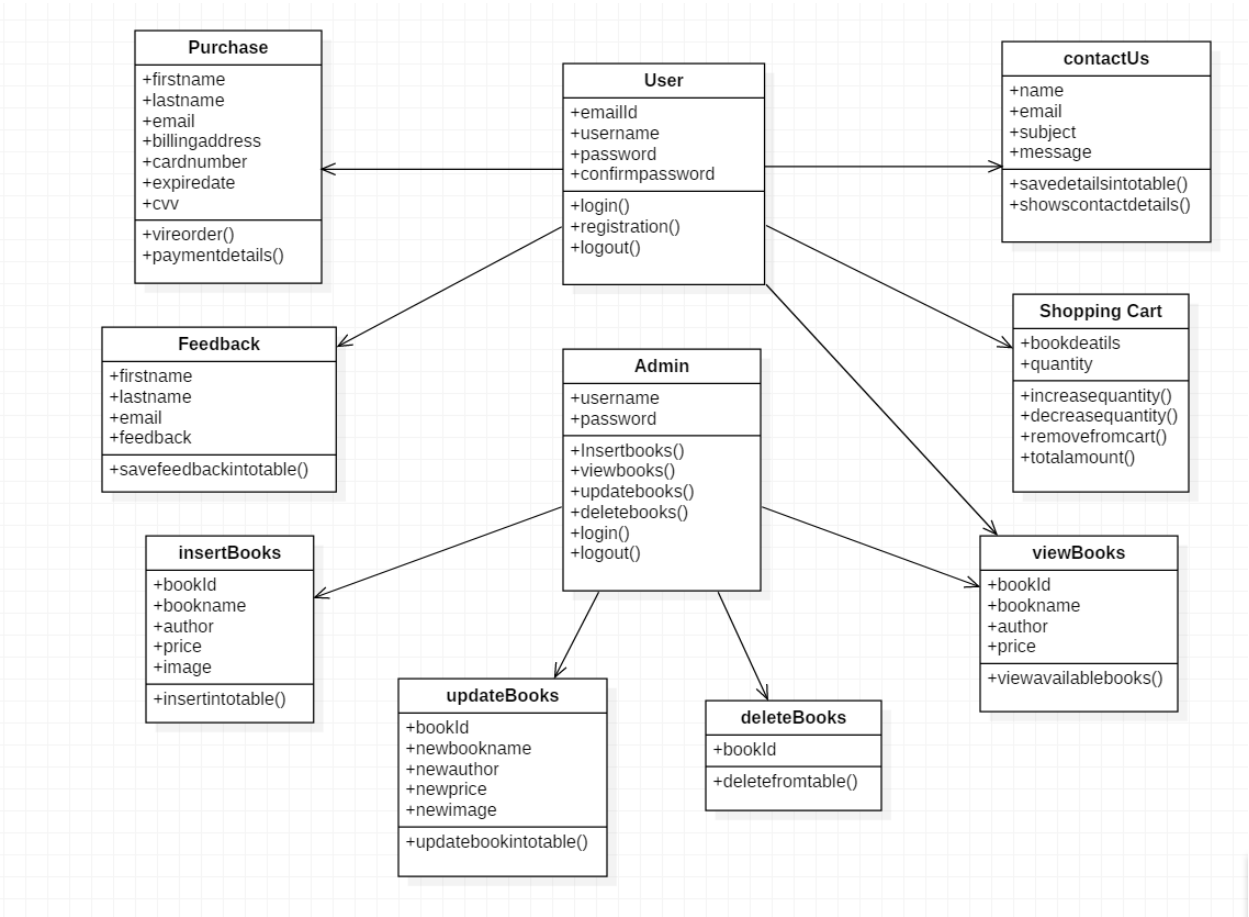


3. Class Diagram

A Class Diagram is a type of static structure diagram in the Unified Modeling Language (UML) that illustrates the structure of a system by showing the classes of objects, their attributes, methods, and relationships among the classes. Class diagrams are widely used in software development to visualize the object-oriented design of a system.

Class diagrams are **useful** for:

Understanding the structure and organization of a system's classes and their relationships.
Communicating the design of a system to stakeholders, including developers, designers, and clients.
Serving as a blueprint for implementing the system in code, especially in object-oriented programming languages like Java, C++, and Python.
Overall, Class Diagrams provide a high-level overview of the static structure of a system, focusing on the classes, their attributes, methods, and relationships, and are an essential tool in software development for designing and documenting object-oriented systems.



4. Use-case Diagram

A use case diagram is a type of behavioral diagram in the Unified Modeling Language (UML) that illustrates the interactions between actors (users or external systems) and a system under consideration. It provides a high-level view of the functionalities or features of a system from the perspective of its users.

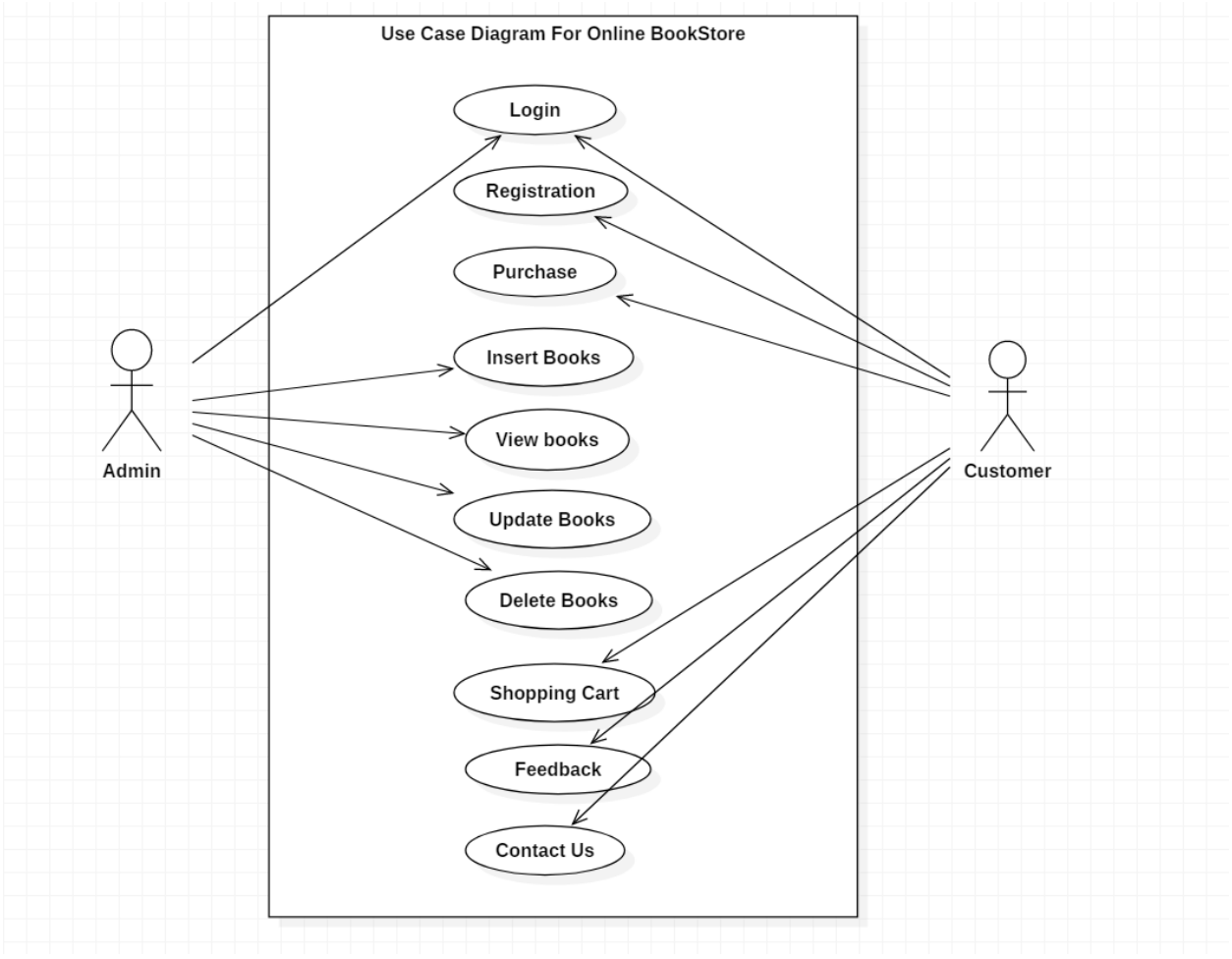
Use case diagrams are commonly used for:

Requirements Analysis: Identifying, documenting, and organizing the functional requirements of a system based on user interactions and goals.

System Design: Providing a visual representation of the system's functionalities and user interactions, aiding in system design and architecture.

Communication: Facilitating communication between stakeholders, including developers, designers, clients, and end-users, by providing a clear and intuitive representation of system functionality.

Overall, use case diagrams are valuable tools for understanding and documenting the behavior of a system from a user's perspective, helping stakeholders align on system requirements and design decisions.



5. Activity Diagram

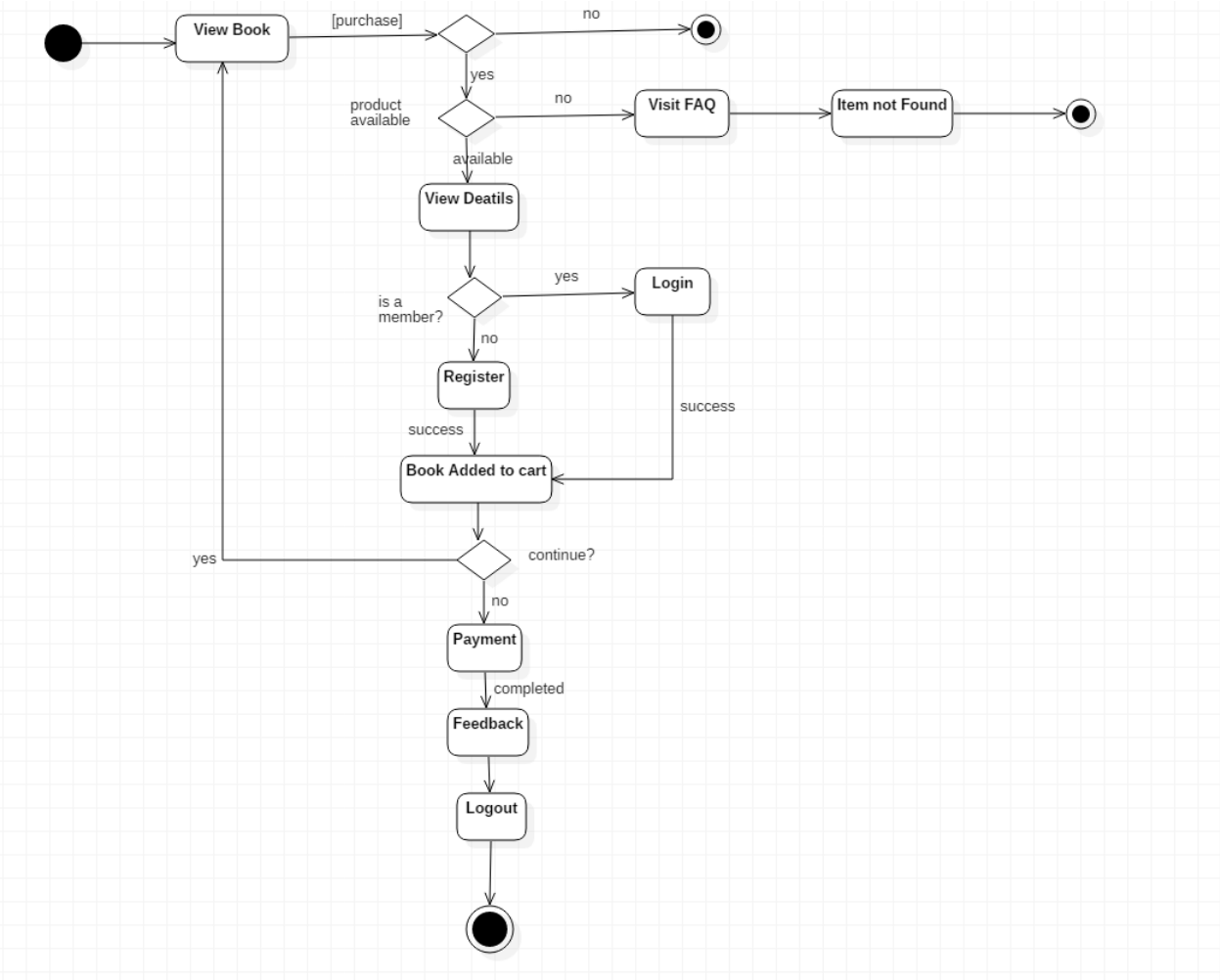
An activity diagram is a behavioral diagram in the Unified Modeling Language (UML) that illustrates the flow of control or behavior of a system or business process. Activity diagrams are used to model the dynamic aspects of a system, showing the sequence of activities, actions, decisions, and transitions between them.

Activity diagrams are commonly used for:

Software System Design: Modeling the behavior of software systems, including the sequence of actions and interactions between components, modules, or subsystems.

Workflow Management: Designing, implementing, and optimizing workflows for various domains, including project management, software development, and manufacturing processes.

Overall, activity diagrams provide a visual representation of the dynamic behavior of a system or process, helping stakeholders understand, analyze, and communicate complex workflows and interactions effectively.



6. Deployment Diagram

A deployment diagram is a type of structural diagram in the Unified Modeling Language (UML) that illustrates the physical deployment of software components and hardware nodes in a distributed system environment. It depicts the configuration of nodes (such as servers, devices, or execution environments) and the relationships between them, showing how software artifacts are deployed on hardware infrastructure.

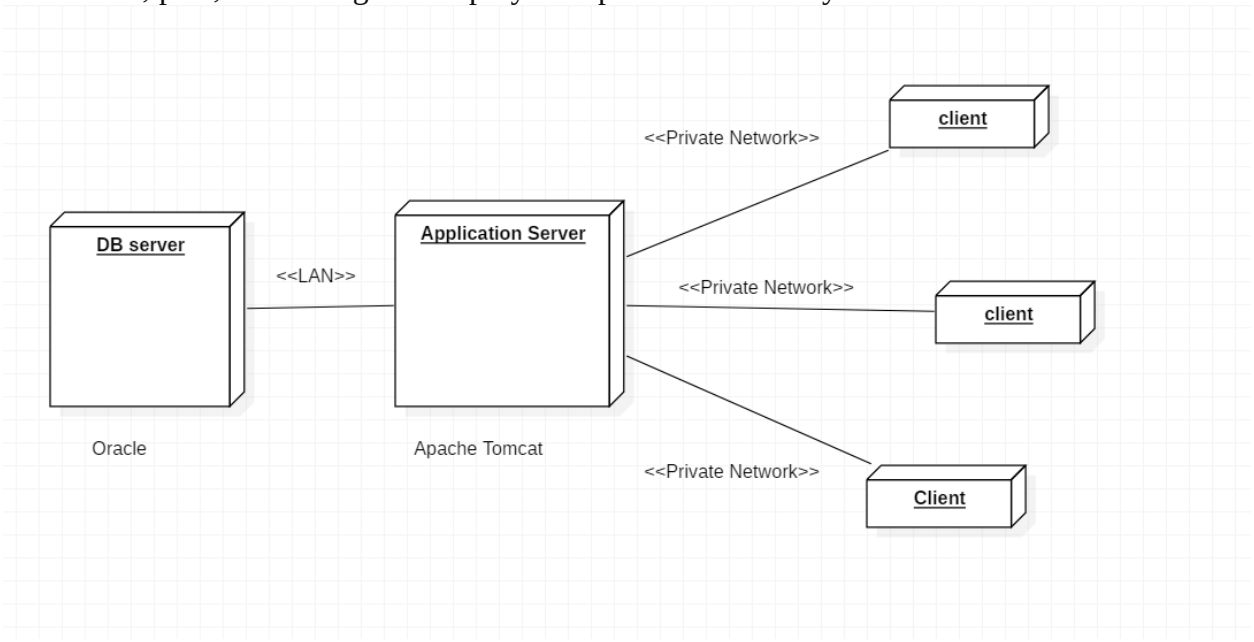
Deployment diagrams are commonly used for:

Visualizing System Architecture: Providing an overview of the physical infrastructure and deployment topology of a distributed system, including servers, databases, and other hardware and software components.

Planning Deployment Strategies: Helping software architects and system administrators plan and organize the deployment of software components across different nodes and environments.

Analyzing Deployment Requirements: Identifying hardware and software resources needed to support the deployment of a system and evaluating scalability, reliability, and performance considerations.

Overall, deployment diagrams provide valuable insights into the physical deployment of software components and hardware infrastructure in a distributed system, helping stakeholders understand, plan, and manage the deployment process effectively.



7. Sequence Diagram

A sequence diagram is a type of behavioral diagram in the Unified Modeling Language (UML) that illustrates the interactions between objects or components in a system over time. It depicts the sequence of messages exchanged between objects, showing the order of execution of these messages and the lifelines of the objects involved.

Sequence diagrams are commonly used for:

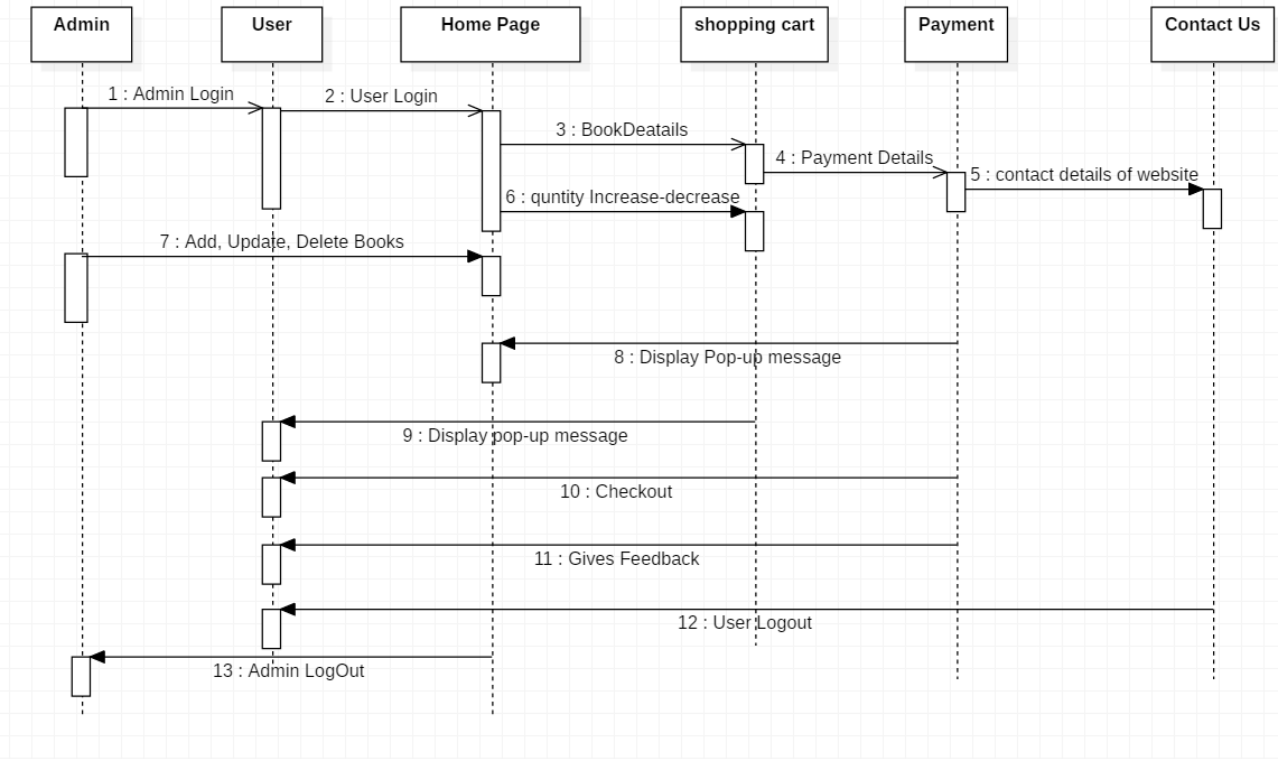
Describing Object Interactions: Visualizing the flow of control and communication between objects or components in a system, helping stakeholders understand the behavior and logic of the system.

Modeling System Behavior: Capturing the dynamic behavior of a system in terms of object interactions and message passing, aiding in system design, analysis, and documentation.

Designing Software Systems: Facilitating the design and implementation of software systems by representing the sequence of method calls, function invocations, and object collaborations in an object-oriented context.

Testing and Debugging: Providing a basis for generating test cases, verifying system behavior, and debugging software applications by simulating object interactions and message exchanges.

Overall, sequence diagrams provide a visual representation of the runtime behavior of a system, focusing on the interactions between objects and the order in which messages are exchanged, helping stakeholders understand and reason about system behavior effectively.



8. Component Diagram

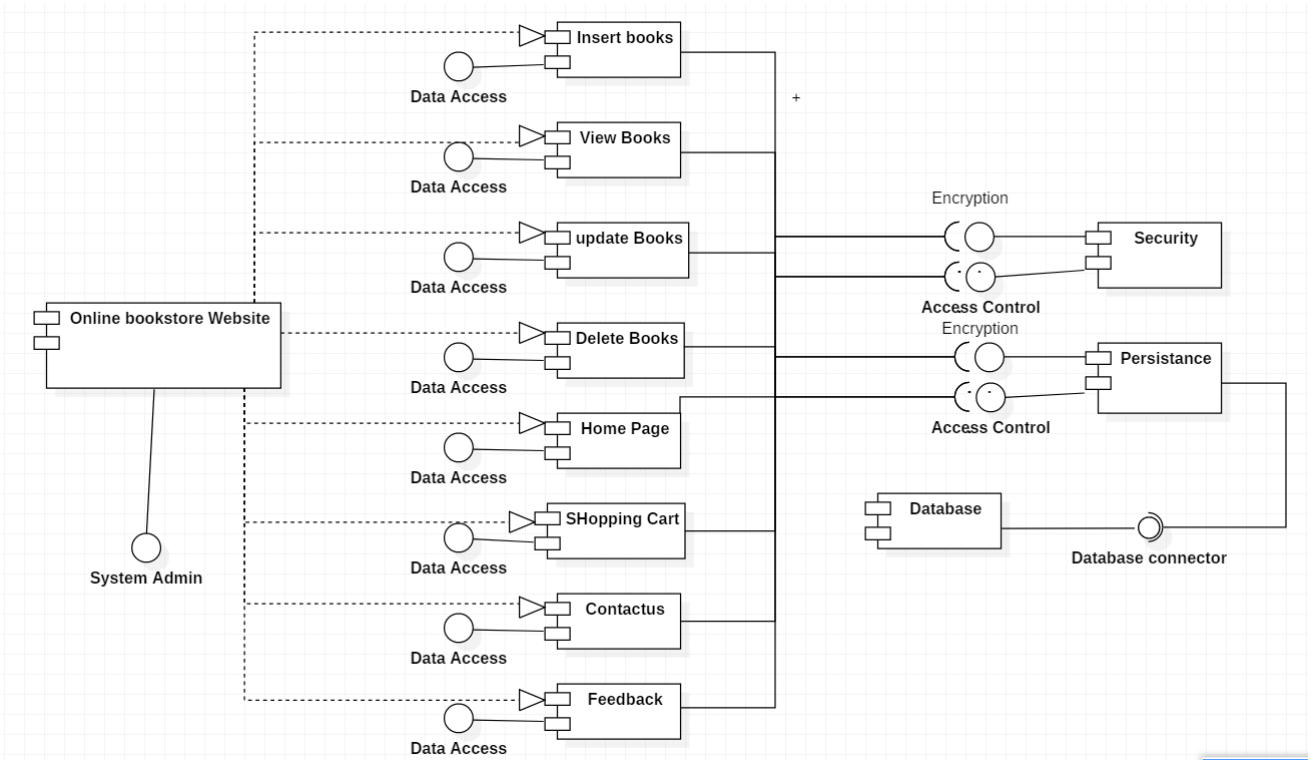
A component diagram in software engineering is a type of diagram that illustrates the components of a system and their dependencies. It's part of the Unified Modeling Language (UML) and is often used during the design phase of software development to visualize the structural relationships between various components within a system.

Component diagrams are used for various purposes in software engineering and system design. **System Design:** Component diagrams are fundamental in the design phase of software development. They help designers and architects break down complex systems into manageable components and illustrate the relationships and dependencies between them.

Communication: Component diagrams serve as a visual communication tool among stakeholders, including developers, architects, project managers, and clients. They provide a clear and concise representation of the system's architecture, facilitating discussions and ensuring everyone involved has a shared understanding of the system's structure.

Modularity and Reusability: By defining clear boundaries between components and their interfaces, component diagrams promote modularity and reusability in software development. Components can be developed, tested, and maintained independently, making it easier to update or replace specific parts of the system without affecting other components.

Overall, component diagrams play a vital role in the software development lifecycle, from the initial design stages through implementation, testing, deployment, and maintenance. They help teams build robust, scalable, and maintainable systems by promoting clear communication, modularity, and dependency management.



9. Object Diagram

An object diagram is a type of structural diagram in the Unified Modeling Language (UML) that represents a snapshot of a system at a specific point in time, showing the objects and their relationships as they exist in memory or at runtime. Object diagrams are used to visualize instances of classes or objects and their associations, attributes, and links at a particular moment during the execution of a system.

Object diagrams are commonly used for:

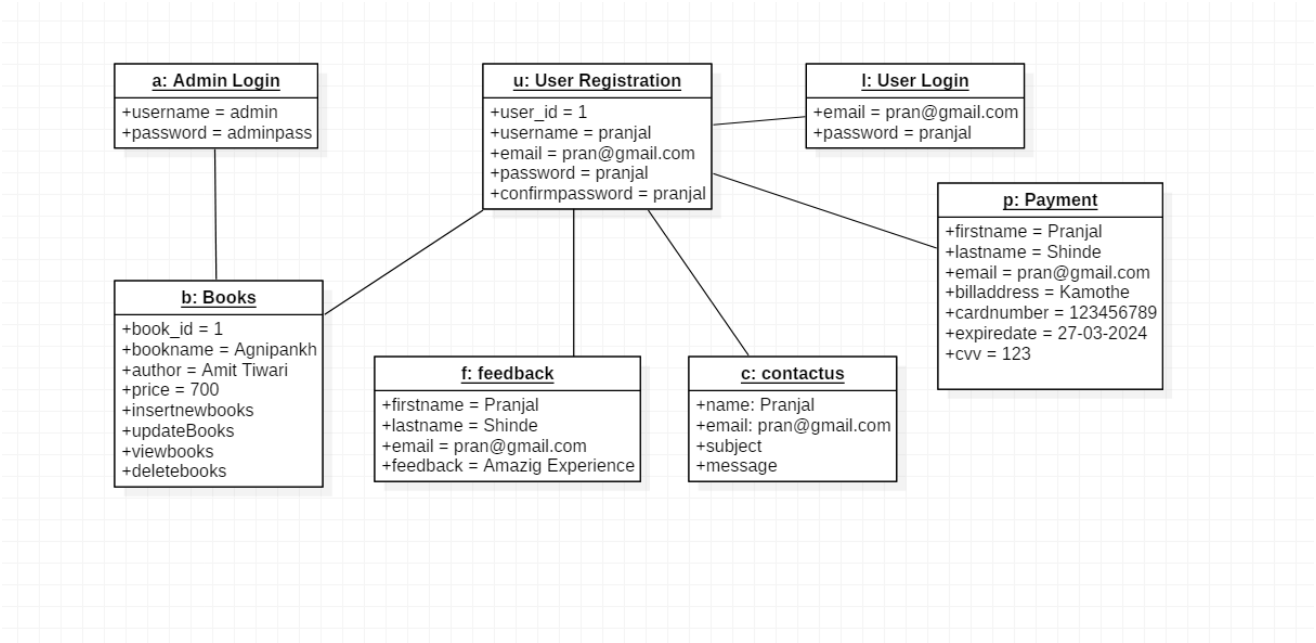
Visualizing Object Instances: Providing a concrete representation of instances of classes or objects and their relationships at a specific point in time, helping stakeholders understand the current state of the system.

Debugging and Testing: Supporting software debugging and testing activities by capturing the runtime state of objects and their interactions, aiding in identifying issues, errors, or inconsistencies within the system.

Designing Object Structures: Facilitating the design and implementation of object-oriented systems by visualizing the structure and relationships of objects, attributes, and associations in a concrete manner.

Communicating System State: Facilitating communication among stakeholders by providing a visual representation of how objects are instantiated and interconnected within the system at a given moment.

Overall, object diagrams provide a snapshot view of the runtime state of a system, focusing on the instances of objects and their relationships as they exist in memory or at a specific point in time. They help stakeholders understand the structure, behavior, and relationships of objects within a system effectively.



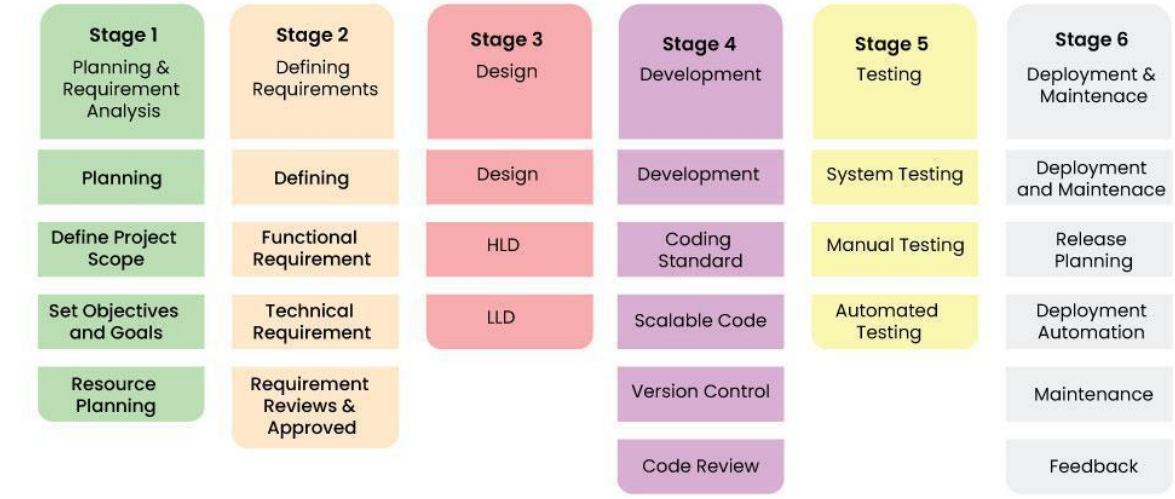
CHAPTER – 5 (SOFTWARE DEVELOPMENT LIFE CYCLE)

5.1 What is SDLC?

Software development life cycle (SDLC) is a structured process that is used to design, develop, and test good-quality software. SDLC, or software development life cycle, is a methodology that defines the entire procedure of software development step-by-step.

SDLC is a process followed for software building within a software organization. SDLC consists of a precise plan that describes how to develop, maintain, replace, and enhance specific software. The life cycle defines a method for improving the quality of software and the all-around development process.

SDLC specifies the task(s) to be performed at various stages by a software engineer or developer. It ensures that the end product is able to meet the customer's expectations and fits within the overall budget. Hence, it's vital for a software developer to have prior knowledge of this software development process.



6 Stages of Software Development Life Cycle

5.2 SDLCS MODEL

To this day, we have more than 50 recognized SDLC models in use. But None of them is perfect, and each brings its favourable aspects and disadvantages for a specific software development project or a team. For this project I'm using Agile Model.

- **Agile Model**

The Agile model is an iterative and incremental approach to software development that emphasizes flexibility, collaboration, and customer feedback. It contrasts with traditional sequential development models, such as the Waterfall model, by breaking the development process into smaller, manageable iterations.

The agile model was mainly designed to adapt to changing requests quickly. The main goal of the Agile model is to facilitate quick project completion. The agile model refers to a group of development processes. These processes have some similar characteristics but also possess certain subtle differences among themselves.

The meaning of Agile is swift or versatile. "Agile process model" refers to a software development approach based on iterative development. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

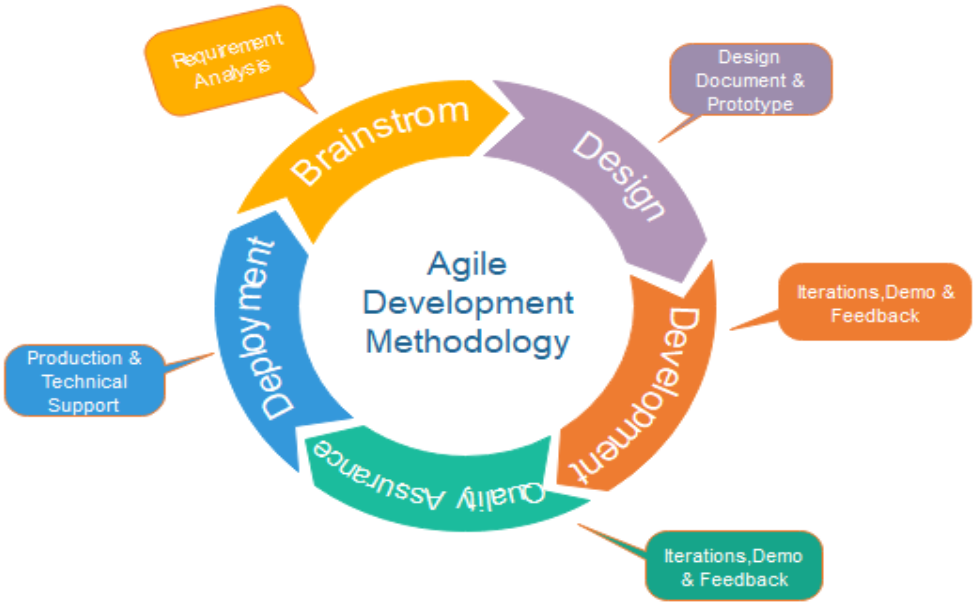


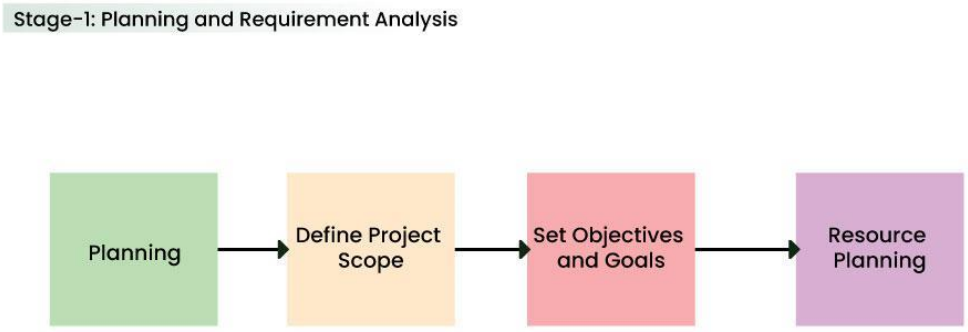
Fig. Agile Model

5.3 PHASES OR STAGES OF SDLC ARE AS FLLOWS:

Stage-1: Planning and Requirement Analysis

Planning is a crucial step in everything, just as in software development. In this same stage, requirement analysis is also performed by the developers of the organization. This is attained from customer inputs, and sales department/market surveys.

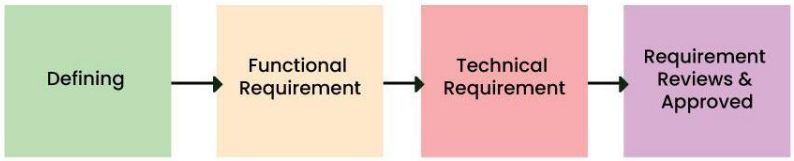
The information from this analysis forms the building blocks of a basic project. The quality of the project is a result of planning. Thus, in this stage, the basic project is designed with all the available information.



Stage-2: Defining Requirements

In this stage, all the requirements for the target software are specified. These requirements get approval from customers, market analysts, and stakeholders.

This is fulfilled by utilizing SRS (Software Requirement Specification). This is a sort of document that specifies all those things that need to be defined and created during the entire project cycle.



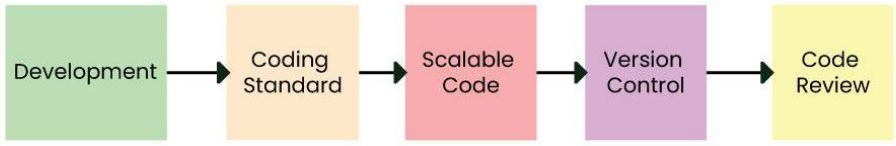
Stage-3: Designing Architecture

SRS is a reference for software designers to come up with the best architecture for the software. Hence, with the requirements defined in SRS, multiple designs for the product architecture are present in the Design Document Specification (DDS).

This DDS is assessed by market analysts and stakeholders. After evaluating all the possible factors, the most practical and logical design is chosen for development.

Stage-4: Developing Website(product)

At this stage, the fundamental development of the product starts. For this, developers use a specific programming code as per the design in the DDS. Hence, it is important for the coders to follow the protocols set by the association. Conventional programming tools like compilers, interpreters, debuggers, etc. are also put into use at this stage. Some popular languages like C/C++, Python, Java, etc. are put into use as per the software regulations.



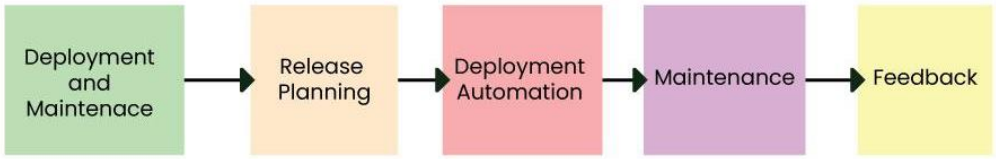
Stage-5: Product Testing and Integration

After the development of the website or product, testing of the software is necessary to ensure its smooth execution. Although, minimal testing is conducted at every stage of SDLC. Therefore, at this stage, all the probable flaws are tracked, fixed, and retested. This ensures that the product confronts the quality requirements of SRS.

Documentation, Training, and Support: Software documentation is an essential part of the software development life cycle. A well-written document acts as a tool and means to information repository necessary to know about software processes, functions, and maintenance. Documentation also provides information about how to use the product. Training in an attempt to improve the current or future employee performance by increasing an employee's ability to work through learning, usually by changing his attitude and developing his skills and understanding.

Stage 6: Deployment and Maintenance of Products

After detailed testing, the conclusive product is released in phases as per the organization's strategy. Then it is tested in a real industrial environment. It is important to ensure its smooth performance. If it performs well, the organization sends out the product as a whole. After retrieving beneficial feedback, the company releases it as it is or with auxiliary improvements to make it further helpful for the customers. However, this alone is not enough. Therefore, along with the deployment, the product's supervision.



Stakeholders, including clients, end-users, project managers, and developers, play crucial roles throughout the SDLC. They contribute to requirements gathering, provide feedback during development, and ensure that the final product aligns with business objectives.

CHAPTER – 6 (DESIGNING)

6.1 SYSTEM DESIGN

System design for this project involves outlining the architecture, components, and functionalities of the system. Here's an overview of the system design for the project:

Architecture:

The system will follow a client-server architecture, where the client-side interface will be accessed through web browsers, and the server-side logic will be implemented using Java Server Pages (JSP) technology.

The web server will host the application and serve web pages to users, while the application server will handle dynamic content generation and business logic processing.

A relational database management system (RDBMS) such as MySQL will store data related to books, users, orders, and other relevant information.

Components:

Client-Side Interface: The user interface (UI) of the website will include pages for browsing books, searching for titles, viewing book details, managing user accounts, and processing orders. It will be implemented using HTML, CSS, Bootstrap and JavaScript.

Server-Side Logic: The server-side logic will be implemented using JSP technology, which allows for dynamic content generation and interaction with the database. Servlets may also be used to handle HTTP requests and responses.

Database: The MySQL database will store various types of data, including book information (title, author, genre, description, etc.), user profiles, order details, and session information.

Middleware: Middleware components may be used to facilitate communication between the client-side interface and server-side logic, ensuring seamless data exchange and processing.

Functionalities:

User Authentication and Authorization: Users will be able to register for accounts, log in securely, and manage their profiles. Authorization mechanisms will control access to certain features based on user roles.

Book Catalog: The website will feature a comprehensive catalog of books, organized by genre, author, popularity, and other criteria. Users can browse, search, and filter books to find titles of interest.

Shopping Cart and Checkout: Users will be able to add books to a shopping cart, review their selections, and proceed to checkout to complete their purchases. Payment processing and order management functionalities will be integrated for a seamless shopping experience.

Admin Panel: An administrative interface will be provided for managing the website, including adding or removing books, managing user accounts, viewing sales reports, and monitoring website performance.

Security:

Security measures will be implemented to protect user data, prevent unauthorized access, and secure online transactions. This includes encryption for sensitive information, secure authentication mechanisms, input validation, and protection against common web vulnerabilities (e.g., SQL injection, cross-site scripting).

By carefully designing the system architecture, components, and functionalities, the "QuillQuest – an Online Bookstore" project can deliver a robust, scalable, and secure platform that meets the

needs and expectations of users while providing a seamless and enjoyable online shopping experience for book enthusiasts.

6.2 DATA DESIGN

USER TABLE:

Attribute Name
User_id
username
email
password
confirmpass

BOOKS TABLE:

Attribute Name
Book_id
bookname
Author
price
images

FEEDBACK TABLE:

Attribute Name
firstname
lastname
email
feedback

CONTACT TABLE:

Attribute Name
name
email
subject
message

PAYMENT TABLE:

Attribute Name
Firstname
Lastname
Email
billaddress
Cardno
expiredate
cvv

6.3DATA INTEGRITY AND CONSTRAINT

USER TABLE:

ATTRIBUTE NAME	DATATYPE	CONTRAINTS
User_id	Int	AUTO_INCREMENT, PRIMARY KEY
username	Varchar(255)	UNIQUE
Email	Varchar(255)	UNIQUE
Password	Varchar(255)	UNIQUE
confirmpass	Varchar(255)	

BOOKS TABLE:

ATTRIBUTE NAME	DATATYPE	CONSTRAINTS
id	Int	PRIMARY KEY, UNIQUE
Bookname	Varchar(255)	UNIQUE
Author	Varchar(255)	
Price	Varchar(255)	
images	Varchar(255)	

FEEDBACK TABLE:

ATTRIBUTE NAME	DATATYPE	CONSTRAINTS
firstname	Varchar(255)	
lstname	Varchar(255)	
email	Varchar(255)	UNIQUE
feedback	Varchar(255)	

CONTACTUS TABLE:

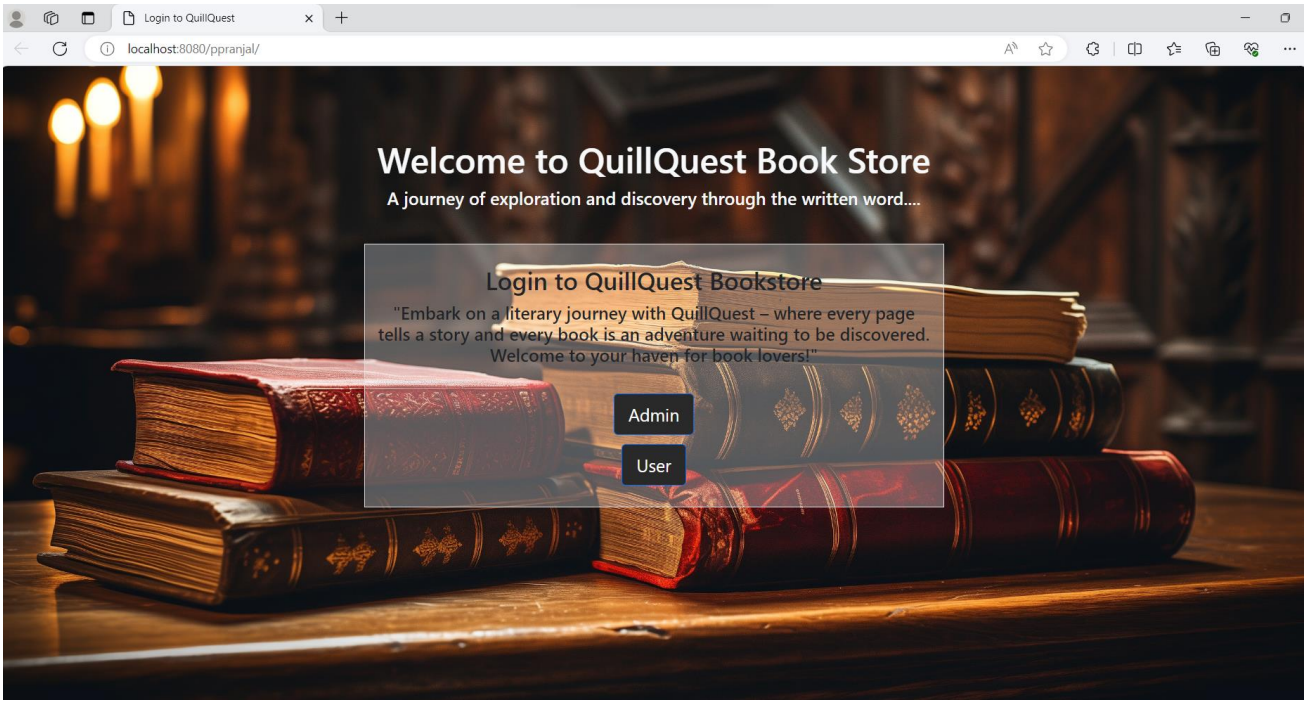
ATTRIBUTE NAME	DATATYPE	CONSTRAINTS
name	Varchar(255)	
email	Varchar(255)	UNNIQUE
subject	Varchar(255)	
message	Varchar(255)	

PAYMENT TABLE:

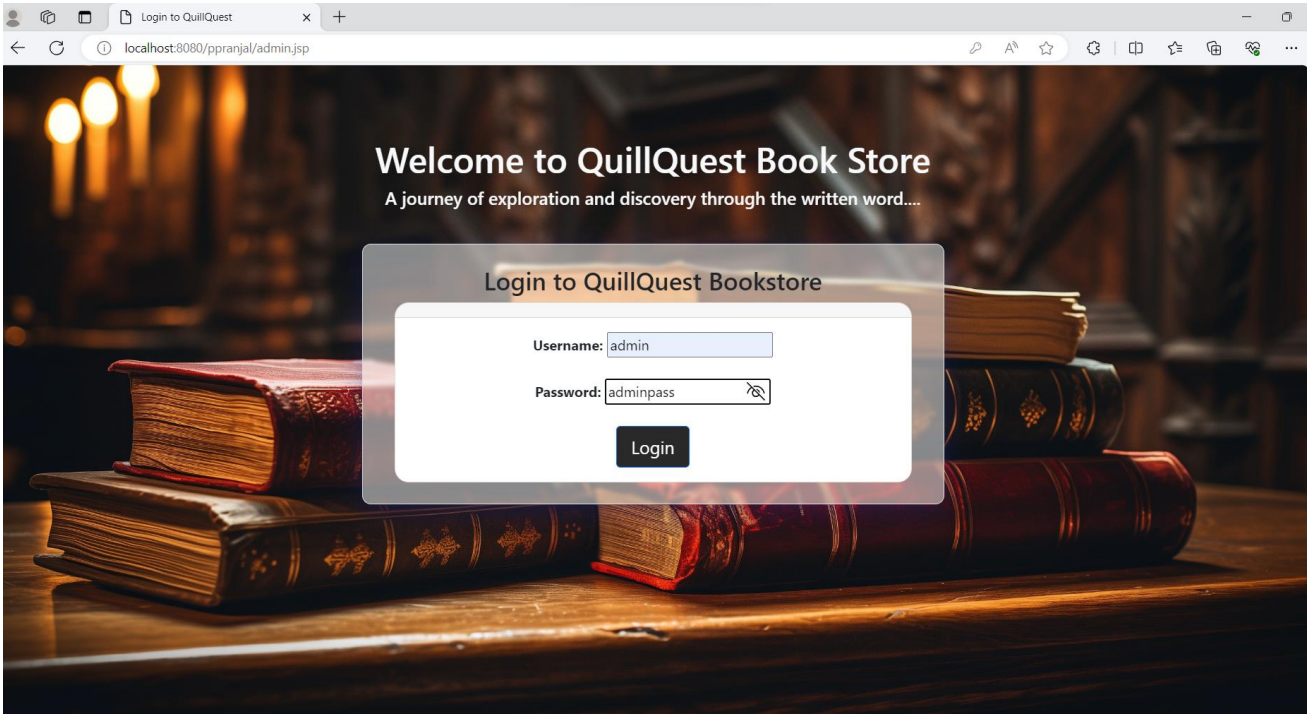
ATTRIBUTE NAME	DATATYPE	CONSTRAINTS
Firstname	Varchar(30)	
Lastanme	Varchar(30)	
Email	Varchar(255)	
Billaddress	Varchar(255)	
cardno	Int	UNIQUE
Expiredate	Varchar(255)	
cvv	Int	UNIQUE

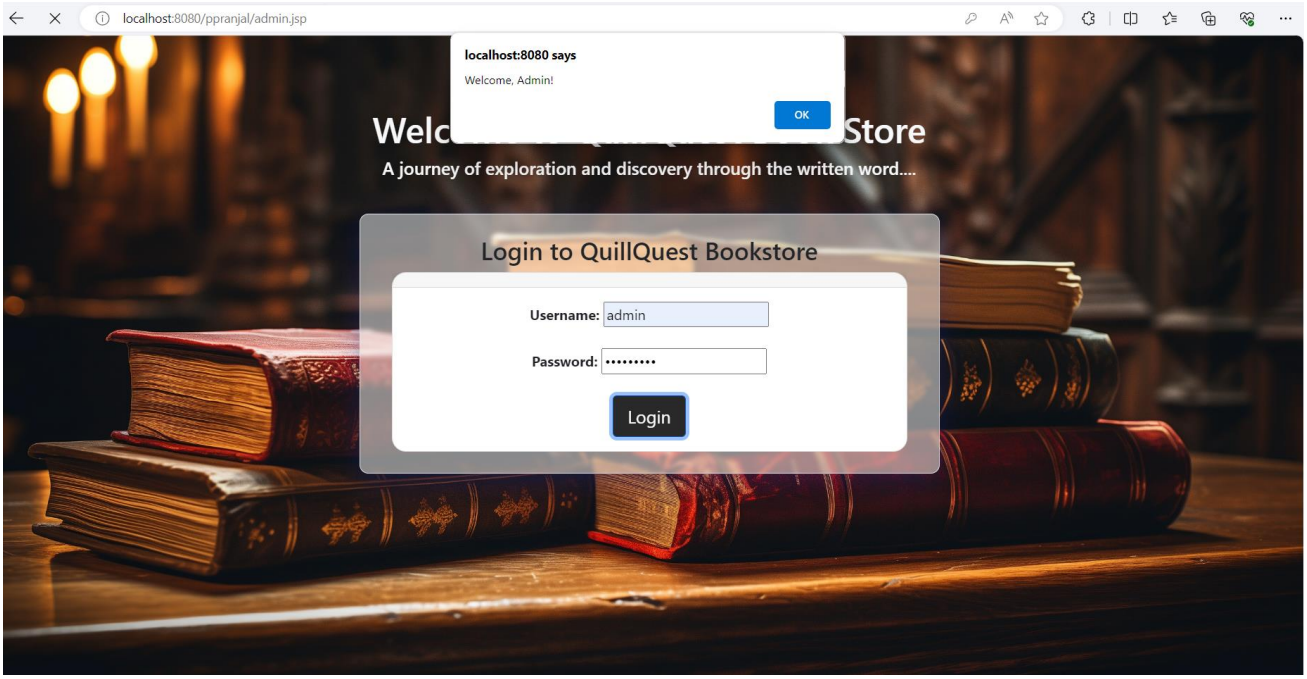
6.4USER INTERFACE AND DESIGN

Main Page:

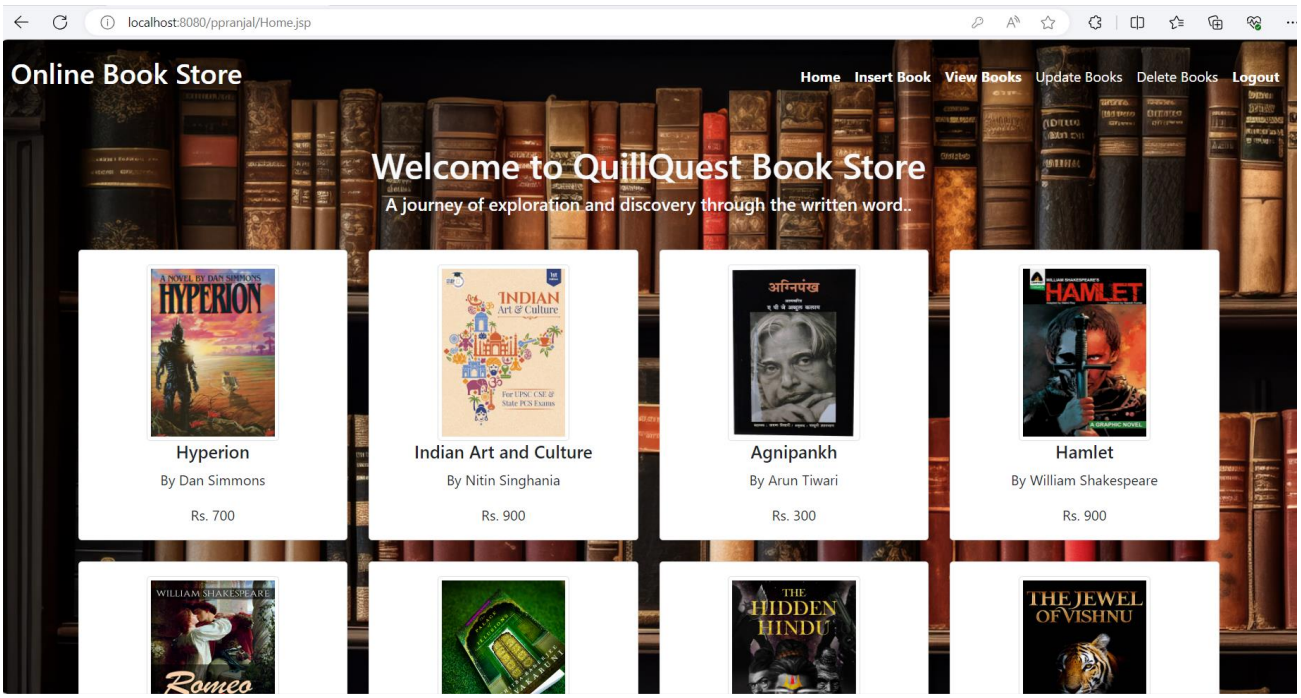


Admin Login:

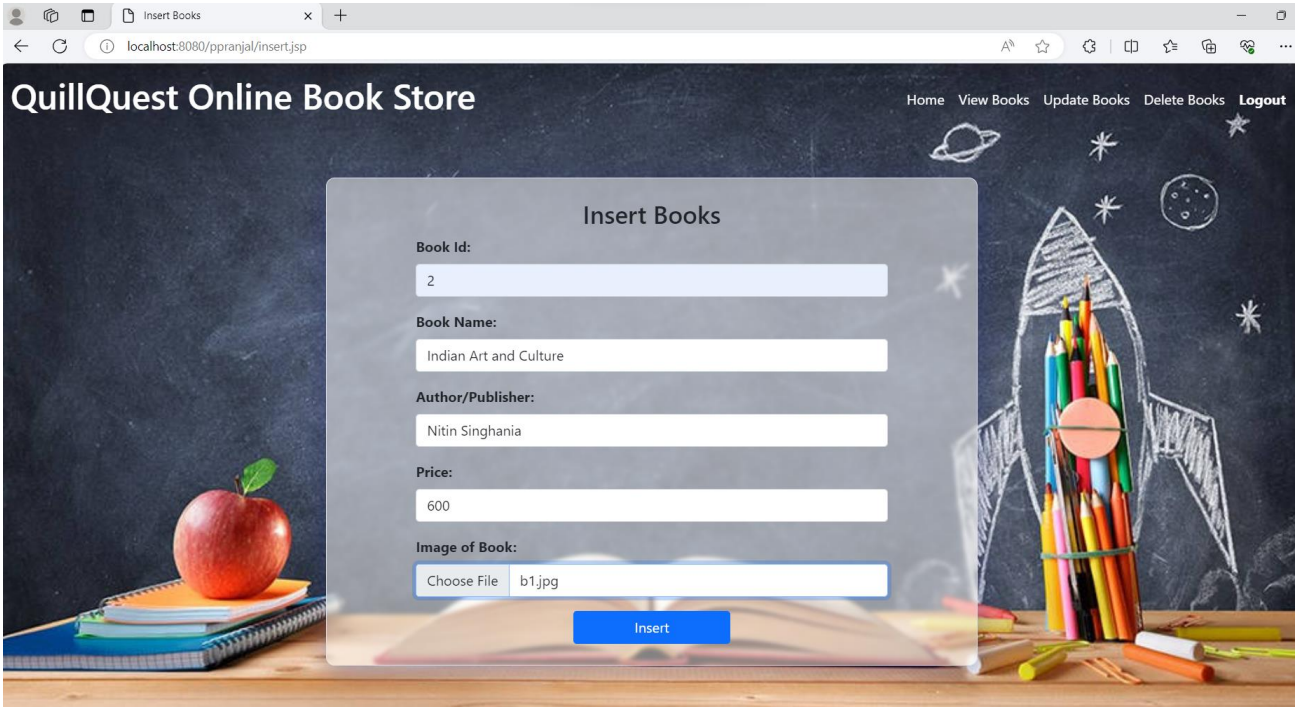




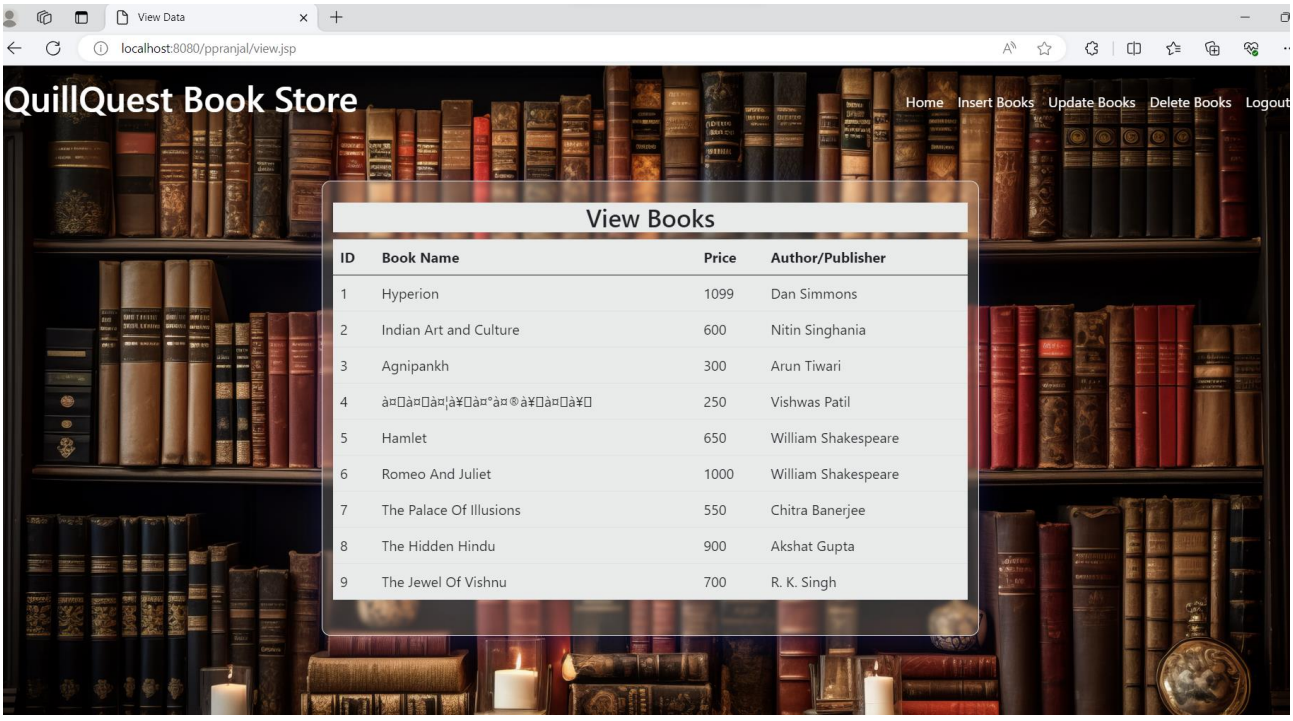
Home Page(Admin Side):



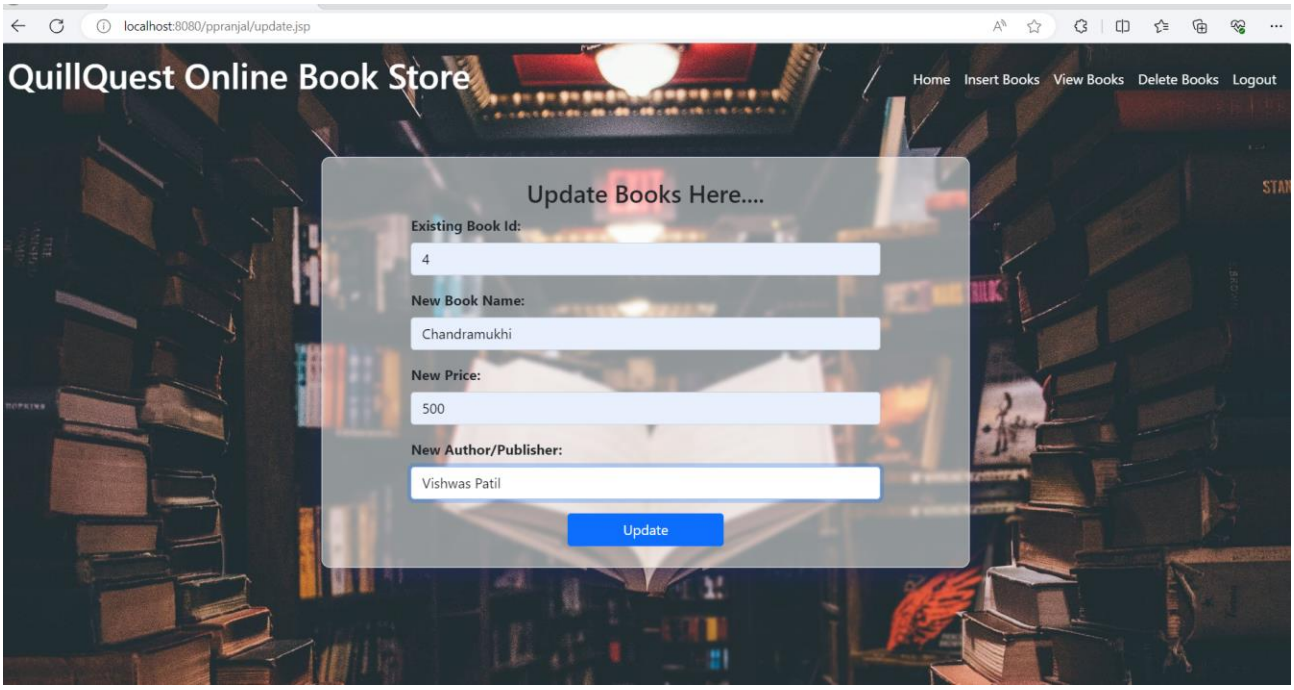
Insert Page:



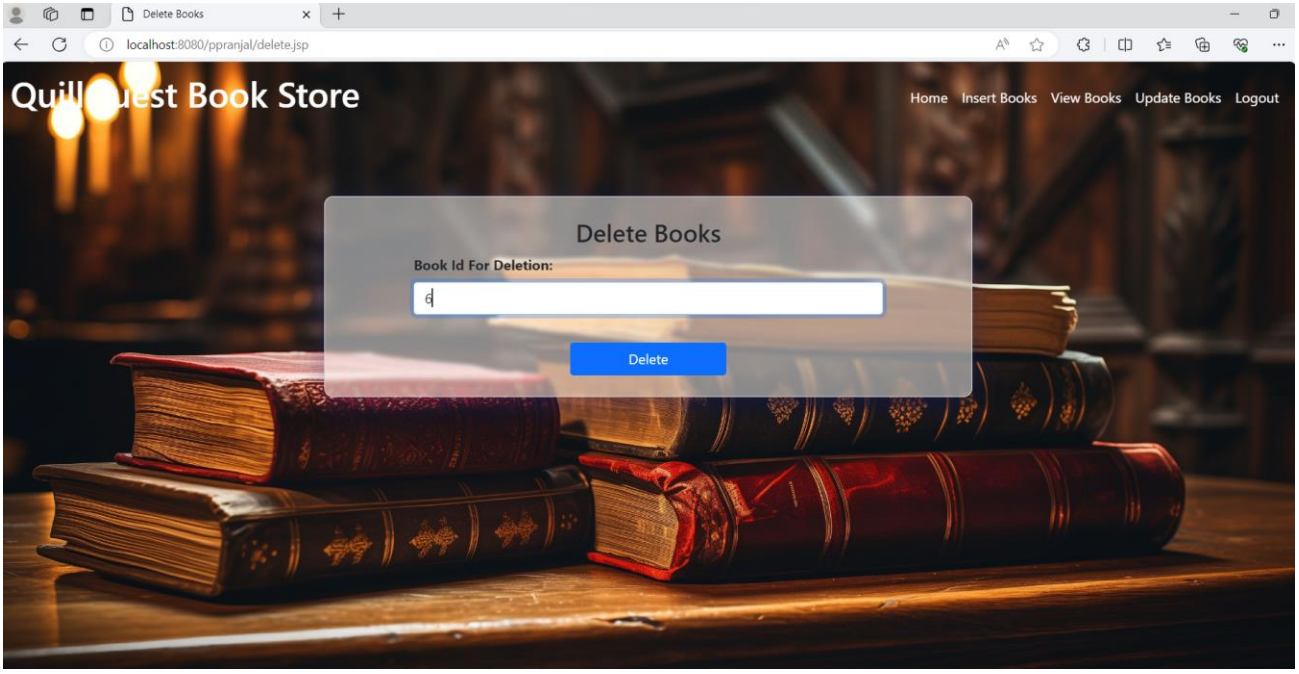
View Page(Admin Side):



Update Book Page:



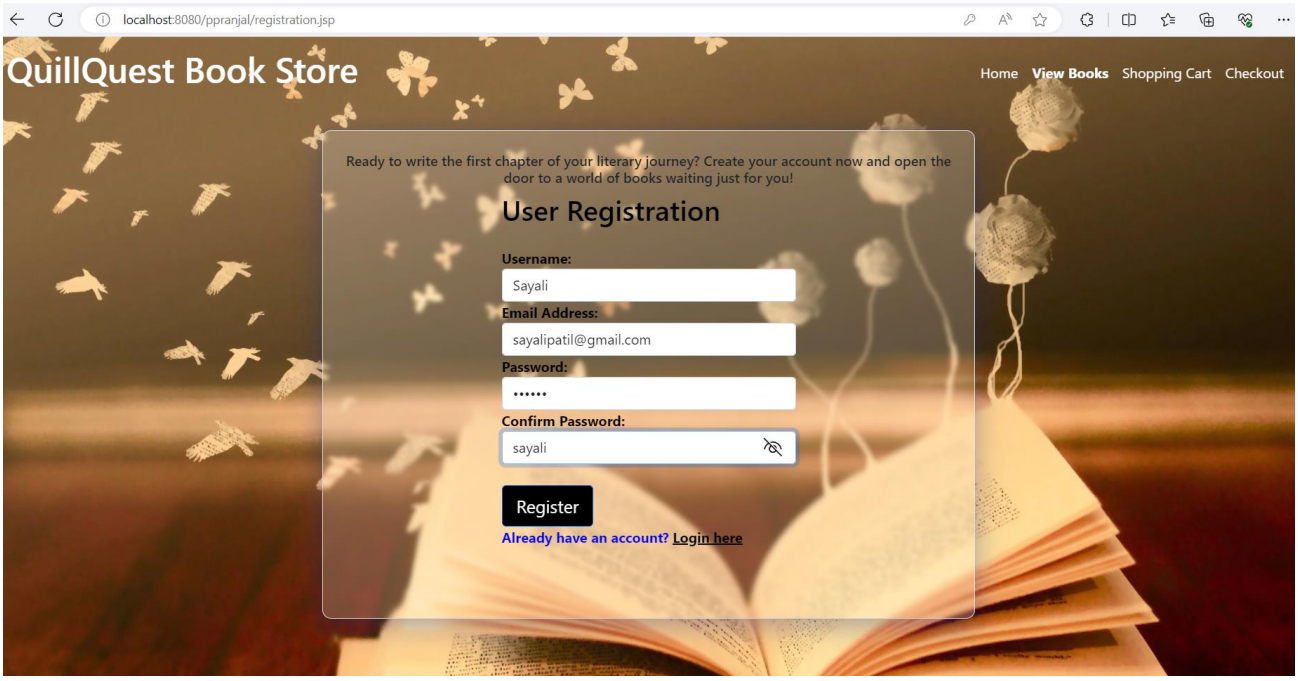
Delete Book Page:



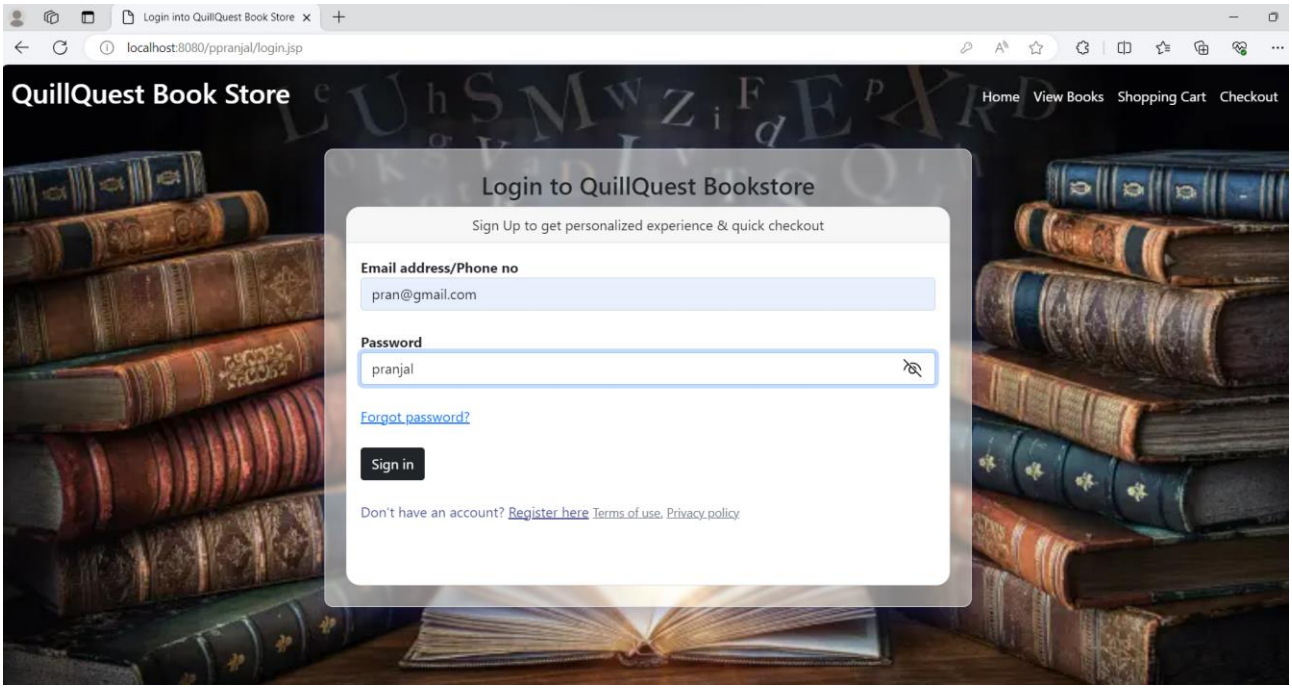
After Click on Logout it will redirect to Main Page.

User Side:

User Registration page:

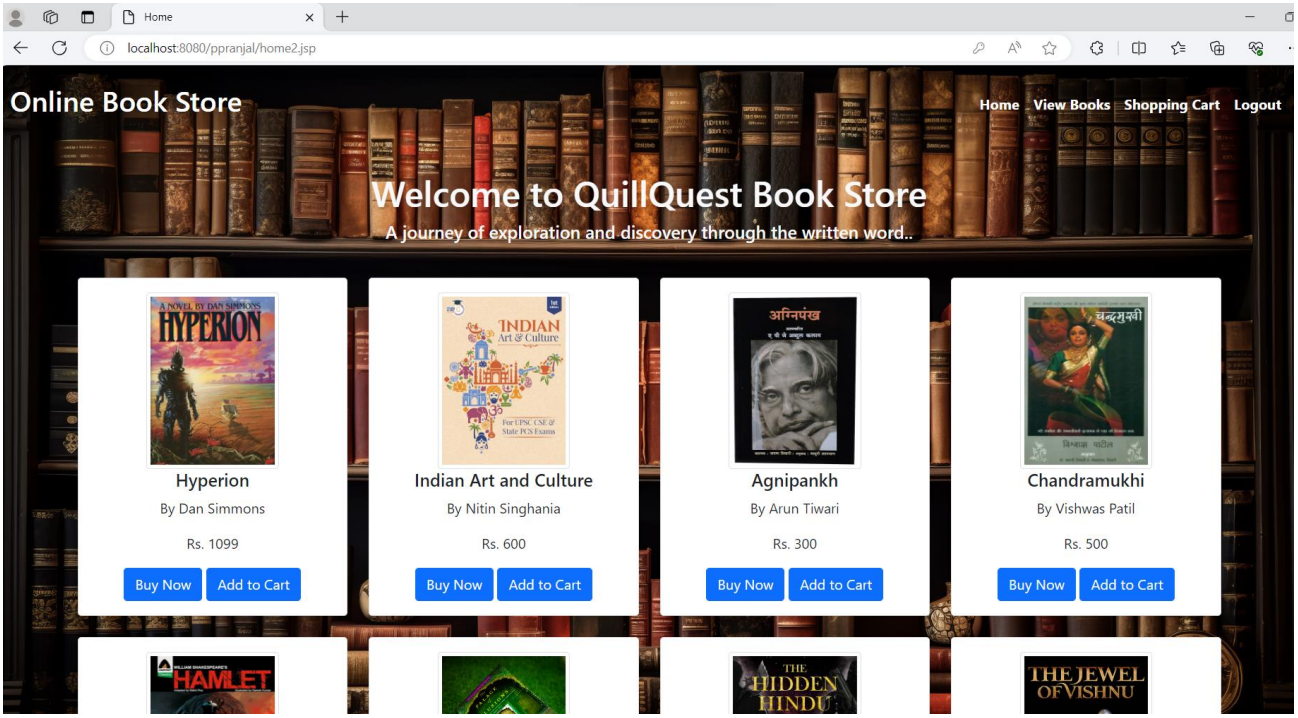


Login Page:

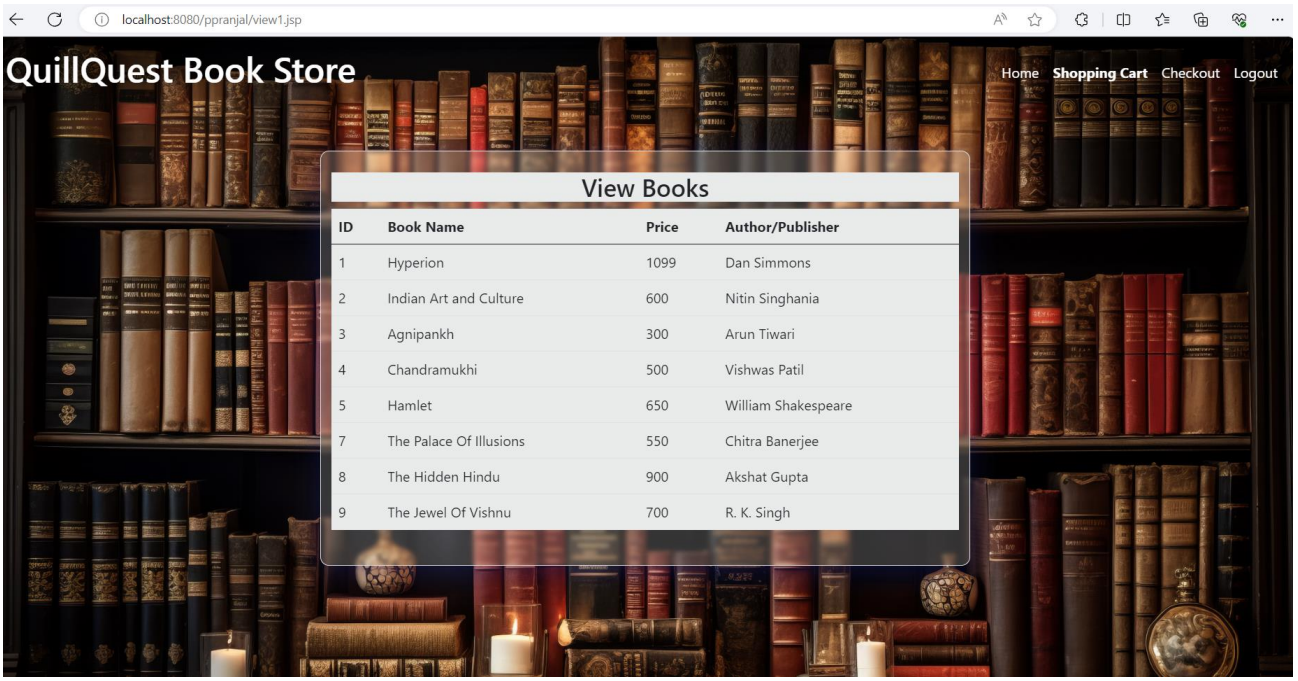


If we click on Forgot Password? It will redirect to main page.

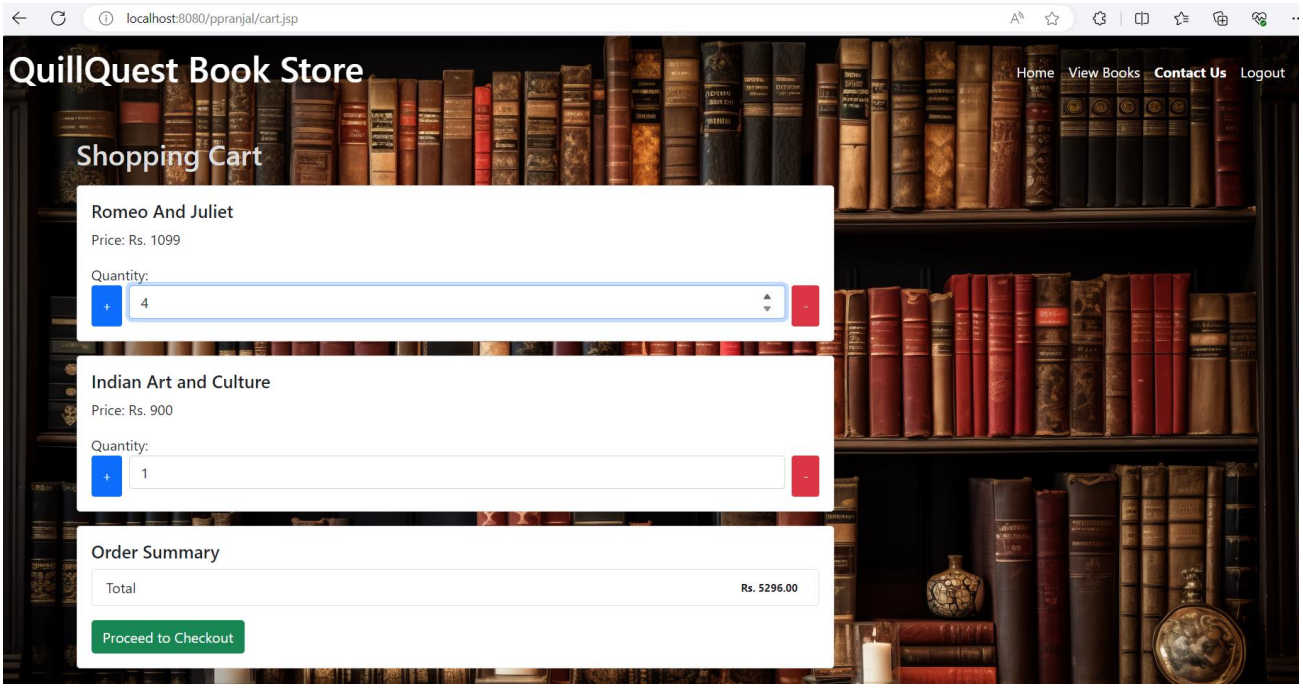
Home Page (User side):



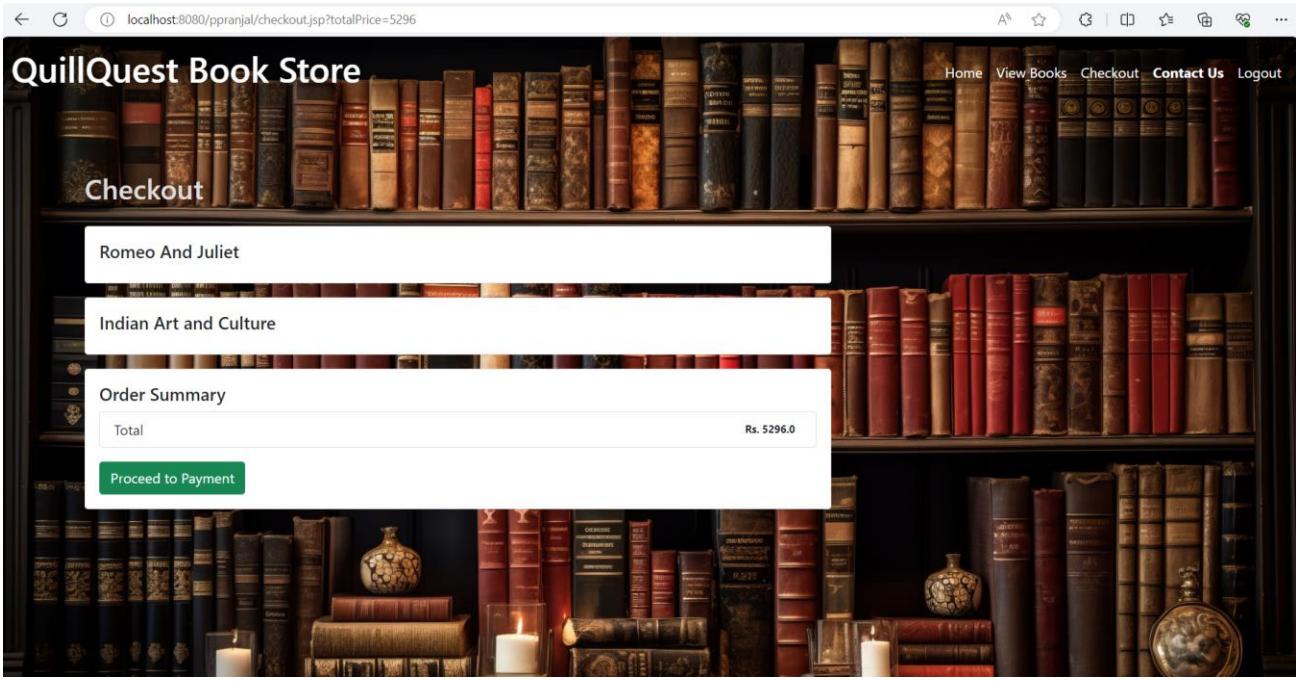
Available Books (User Side):



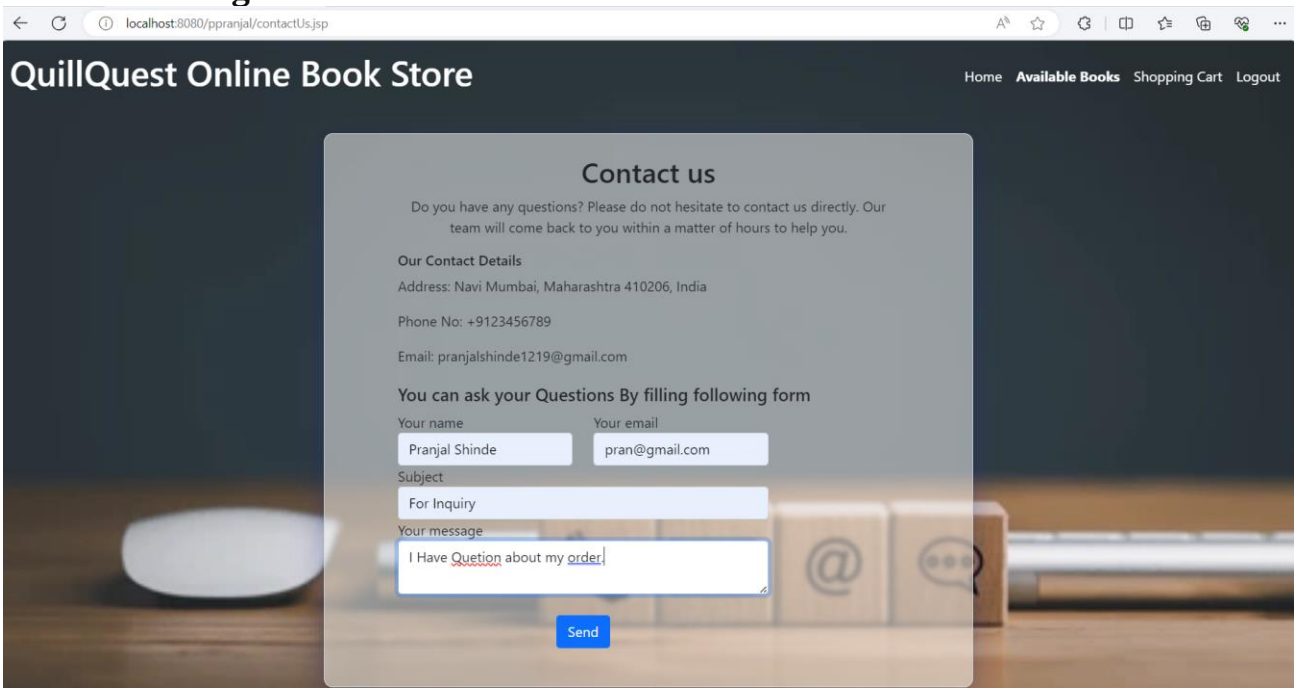
Shopping-Cart Page:



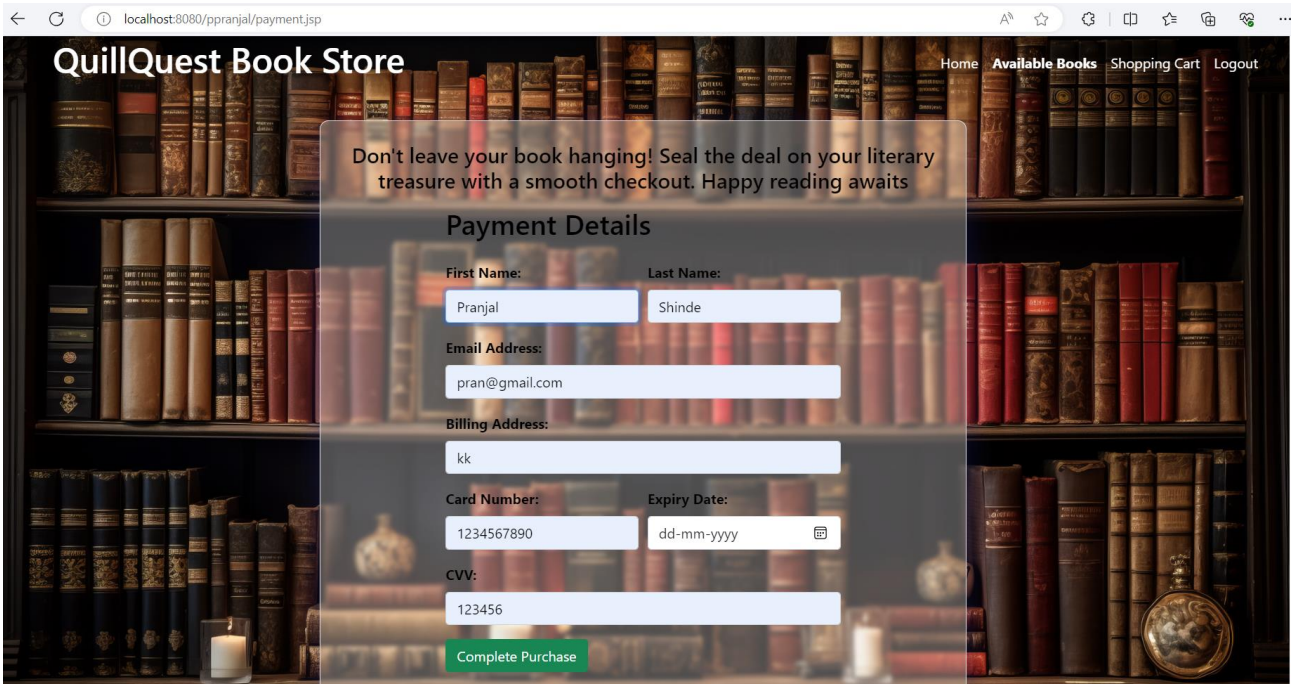
Checkout Page:



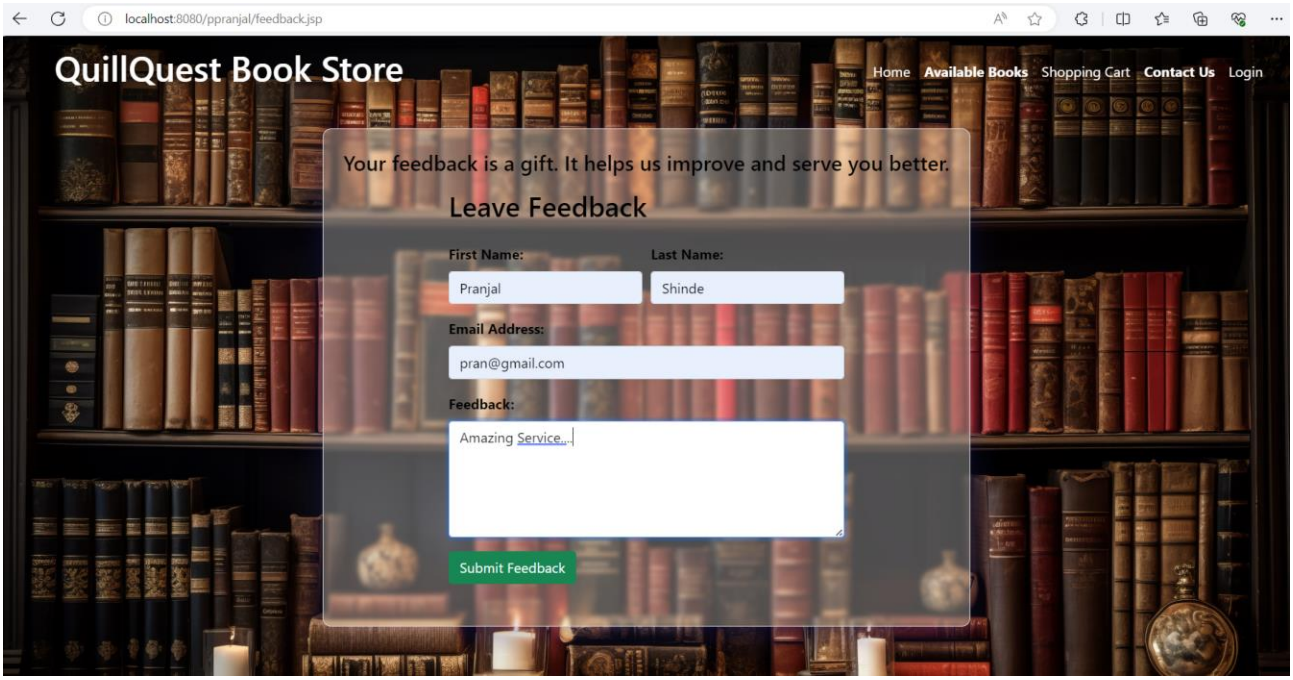
Contact Us Page:



Payment Page:

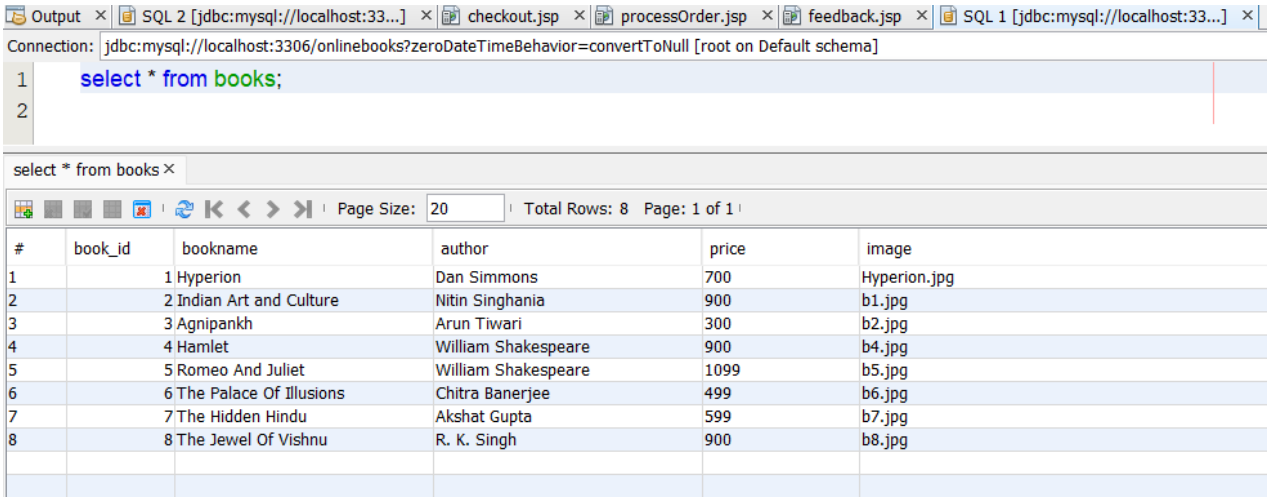


Feedback Form:



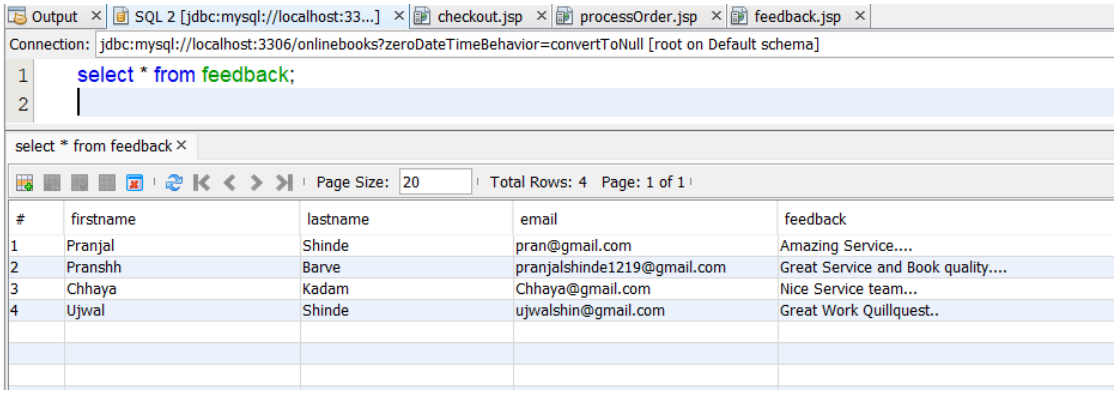
Database Views:

Books Table:



#	book_id	bookname	author	price	image
1	1	Hyperion	Dan Simmons	700	Hyperion.jpg
2	2	Indian Art and Culture	Nitin Singhania	900	b1.jpg
3	3	Agnipankh	Arun Tiwari	300	b2.jpg
4	4	Hamlet	William Shakespeare	900	b4.jpg
5	5	Romeo And Juliet	William Shakespeare	1099	b5.jpg
6	6	The Palace Of Illusions	Chitra Banerjee	499	b6.jpg
7	7	The Hidden Hindu	Akshat Gupta	599	b7.jpg
8	8	The Jewel Of Vishnu	R. K. Singh	900	b8.jpg

Feedback Table:



Output | SQL 2 [jdbc:mysql://localhost:33...] | checkout.jsp | processOrder.jsp | feedback.jsp

Connection: jdbc:mysql://localhost:3306/onlinebooks?zeroDateTimeBehavior=convertToNull [root on Default schema]

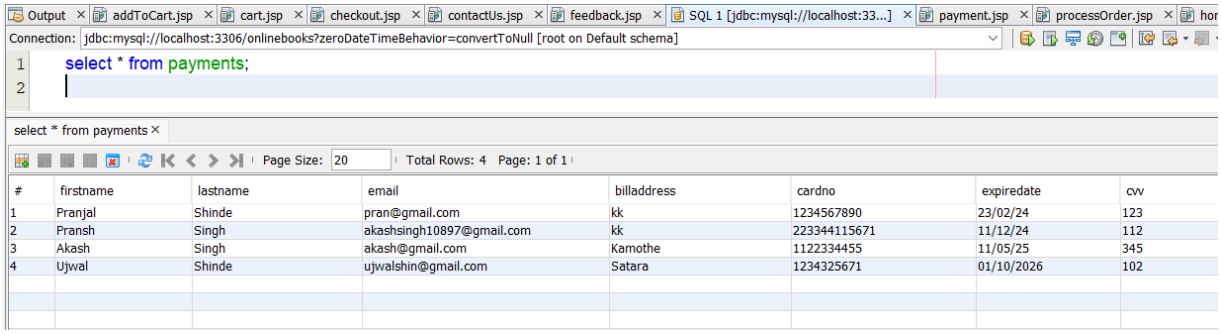
```
1  select * from feedback;
2
```

select * from feedback

Page Size: 20 Total Rows: 4 Page: 1 of 1

#	firstname	lastname	email	feedback
1	Pranjal	Shinde	pran@gmail.com	Amazing Service....
2	Pranshh	Barve	pranjalshinde1219@gmail.com	Great Service and Book quality....
3	Chhaya	Kadam	Chhaya@gmail.com	Nice Service team...
4	Ujwal	Shinde	ujwalshin@gmail.com	Great Work Quillquest..

Payment Table:



Output | addToCart.jsp | cart.jsp | checkout.jsp | contactUs.jsp | feedback.jsp | SQL 1 [jdbc:mysql://localhost:33...] | payment.jsp | processOrder.jsp

Connection: jdbc:mysql://localhost:3306/onlinebooks?zeroDateTimeBehavior=convertToNull [root on Default schema]

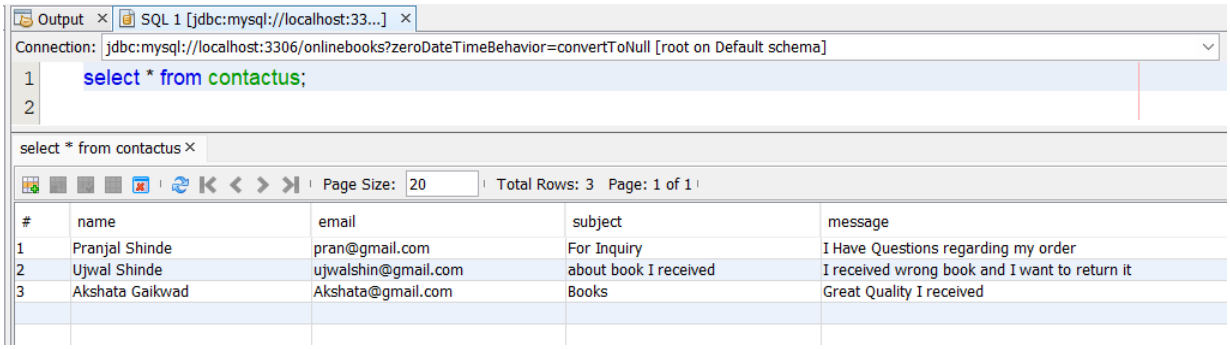
```
1  select * from payments;
2
```

select * from payments

Page Size: 20 Total Rows: 4 Page: 1 of 1

#	firstname	lastname	email	billaddress	cardno	expiredate	cvv
1	Pranjal	Shinde	pran@gmail.com	kk	1234567890	23/02/24	123
2	Pransh	Singh	akashsingh10897@gmail.com	kk	223344115671	11/12/24	112
3	Akash	Singh	akash@gmail.com	Kamothe	1122334455	11/05/25	345
4	Ujwal	Shinde	ujwalshin@gmail.com	Satara	1234325671	01/10/2026	102

ContactUs Table:



Output | SQL 1 [jdbc:mysql://localhost:33...]

Connection: jdbc:mysql://localhost:3306/onlinebooks?zeroDateTimeBehavior=convertToNull [root on Default schema]

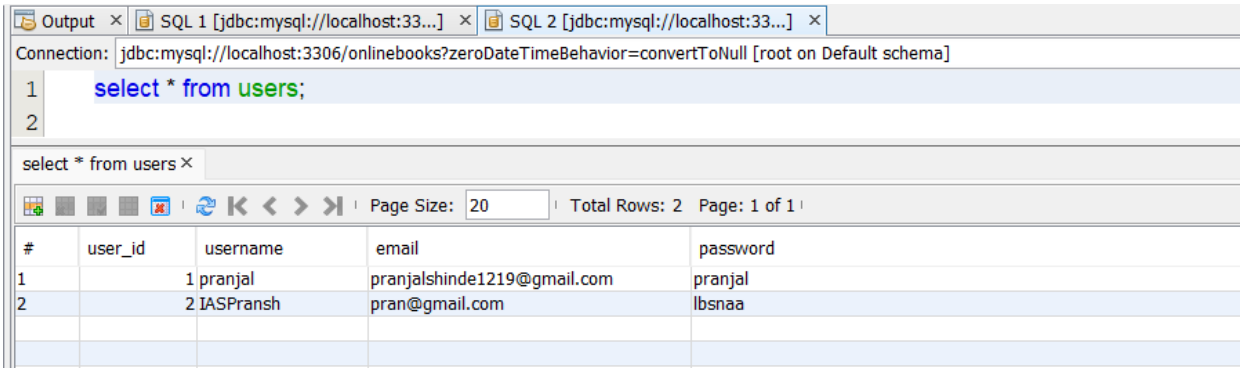
```
1  select * from contactus;
2
```

select * from contactus

Page Size: 20 Total Rows: 3 Page: 1 of 1

#	name	email	subject	message
1	Pranjal Shinde	pran@gmail.com	For Inquiry	I Have Questions regarding my order
2	Ujwal Shinde	ujwalshin@gmail.com	about book I received	I received wrong book and I want to return it
3	Akshata Gaikwad	Akshata@gmail.com	Books	Great Quality I received

Users Table:



Output | SQL 1 [jdbc:mysql://localhost:33...] | SQL 2 [jdbc:mysql://localhost:33...]

Connection: jdbc:mysql://localhost:3306/onlinebooks?zeroDateTimeBehavior=convertToNull [root on Default schema]

```
1  select * from users;
2
```

select * from users

Page Size: 20 Total Rows: 2 Page: 1 of 1

#	user_id	username	email	password
1	1	pranjal	pranjalshinde1219@gmail.com	pranjal
2	2	IASPransh	pran@gmail.com	lbsnaa

CHAPTER– 7 (CODING / IMPLEMENTATION)

7.1 CODING

DatabaseUtils.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="java.sql.*"%>
<%@ page import="java.io.*" %>
<%@ page import="javax.servlet.http.*, javax.servlet.*" %>
<%@ page import="javax.servlet.annotation.MultipartConfig" %>

<!DOCTYPE html>
<%!
   static Connection getConnection() throws SQLException, ClassNotFoundException{

      String url = "jdbc:mysql://localhost:3306/onlinebooks";
      String username = "root";
      String password = "";

      Class.forName("com.mysql.jdbc.Driver");
      return DriverManager.getConnection(url, username, password);
   }
%>
```

Home.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@include file="DatabaseUtils.jsp" %>
<!DOCTYPE html>
<html>
<head>
   <meta charset="UTF-8">
   <title>Home</title>
   <style>
      .crd-ho:hover{
         background-color: #ebd9b4;
      }
   </style>
   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
</head>
<body style="background-image:url('https://images4.alphacoders.com/132/1326368.png') ;
height: 100vh; background-size: auto;">
<nav class="navbar navbar-expand-lg navbar-dark bg-#cfa1ba py-3">
   <div class="container-fluid pl-5">
      <a class="navbar-brand" href="#"><h2>Online Book Store</h2></a>
```

```
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
         <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarSupportedContent">
         <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
            <li class="nav-item">
               <a class="nav-link active" aria-current="page"
href="index.jsp"><b>Home</b></a>
            </li>
            <li class="nav-item">
               <a class="nav-link active" aria-current="page" href="insert.jsp"><b>Insert
Book</b></a>
            </li>
            <li class="nav-item">
               <a class="nav-link active" aria-current="page" href="view.jsp"><b>View
Books</b></a>
            </li>
            <li class="nav-item">
               <a class="nav-link active" aria-current="page" href="update.jsp">Update
Books</a>
            </li>
            <li class="nav-item">
               <a class="nav-link active " aria-current="page" href="delete.jsp">Delete Books</a>
            </li>
            <li class="nav-item">
               <a class="nav-link active" aria-current="page"
href="index.jsp"><b>Logout</b></a>
            </li>
         </ul>
      </div>
   </div>
</nav><br>
<div class="jumbotron jumbotron-fluid text-center bg-#88AB8E  text-light py-2">
   <div class="container">
      <h1 class="text:center">Welcome to  QuillQuest  Book Store </h1>
      <h5 class="text:center">A journey of exploration and discovery through the written word..
</h5>
   </div>
</div><br>
<div class="container text-center">
   <div class="row row-cols-1 row-cols-md-4 g-4">
      <!-- Java code to fetch books from the database and dynamically generate cards for each
book -->
      <%
```

```
        Connection conn = null;
        PreparedStatement stmt = null;
        ResultSet rs = null;
        try {
            conn = getConnection(); // Get connection from DatabaseUtils.jsp
            String sql = "SELECT * FROM books";
            stmt = conn.prepareStatement(sql);
            rs = stmt.executeQuery();

            while (rs.next()) {
                String bookTitle = rs.getString("bookname");
                String author = rs.getString("author");

                String price = rs.getString("price");
    %>
    <div class="col-md-3">
        <div class="card crd-ho">
            <div class="card-body text-center">
                <img alt="<%= bookTitle %>" src="images/<%out.println(rs.getString(5));%>"
style="width:150px; height:200px" class="img-thumbnail">
                <h5 class="card-title"><%= bookTitle %></h5>
                <p class="card-text">By <%= author %></p>
                <p class="card-text">Rs. <%= price %></p>
            </div>
        </div>
    </div>
    <%
            }
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            // Close resources
            if (rs != null) { try { rs.close(); } catch (SQLException e) { e.printStackTrace(); } }
            if (stmt != null) { try { stmt.close(); } catch (SQLException e) { e.printStackTrace(); }
}
            if (conn != null) { try { conn.close(); } catch (SQLException e) { e.printStackTrace();
} }
        }
    %>
  </div>
</div>
<br>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js"
integrity="sha384-
cVKIPhGWiC2Al4u+LWgxfKTRIcfu0JTxR+EQDz/bgldoEyl4H0zUF0QKbrJ0EcQF"
crossorigin="anonymous"></script>
</body>
</html>
```

Admin.jsp

```
<!-- <%@page contentType="text/html" pageEncoding="UTF-8"%> -->
<%@include file="DatabaseUtils.jsp" %>
<!DOCTYPE html>
<html>
<head>
   <meta charset="UTF-8">
   <title>Login to QuillQuest</title>
   <style>
      .crd-ho:hover{
         background-color: #ebd9b4;
      }
   </style>
   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
</head>
<body style="background-image:url('https://images5.alphacoders.com/133/1338186.png') ;
height: 100vh; background-size: cover;">
  <nav class="navbar navbar-expand-lg navbar-dark bg-#cfa1ba py-3">
   <div class="container-fluid pl-5">
     <a class="navbar-brand" href="#"><h2></h2></a>
     <button class="navbar-toggler" type="button" data-b   s-toggle="collapse" data-bs-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
       <span class="navbar-toggler-icon"></span>
     </button>
     <div class="collapse navbar-collapse" id="navbarSupportedContent">
       <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
       </ul>
     </div>
   </div>
  </nav><br>
  <div class="jumbotron jumbotron-fluid text-center bg-#88AB8E  text-light py-2">
   <div class="container">
     <h1 class="text:center">Welcome to  QuillQuest  Book Store </h1>
     <h5 class="text:center">A journey of exploration and discovery through the written word....
</h5>
   </div>
  </div><br>
  <div class="container text-center" >
     <div class="row row-cols-1 row-cols-md-4 g-4">
  <div class="container border py-4 w-50" style="background: rgba( 255, 255, 255, 0.5 );
  box-shadow: 0 8px 32px 0 rgba( 31, 38, 135, 0.37 );
  backdrop-filter: blur( 2.5px );
```

```
  -webkit-backdrop-filter: blur( 2.5px );
  border-radius: 10px;
  border: 1px solid rgba(255, 255, 255, 0.18 );">
    <h3 class="text-center">Login to QuillQuest Bookstore</h3>
   <form id="loginForm" method="post"  style="display: grid;place-items: center;">
<body style="background-color:rgb(151, 248, 254);">
<div class="container">
    <div class="card" style="border-radius: 1rem;">
        <div class="card-header text-center"></div>
        <div class="card-body">
            <label for="username"><b>Username:</b></label>
            <input type="text" id="username" name="username" required>
            <br>
            <br>
            <label for="password"><b>Password:</b></label>
            <input type="password" id="password" name="password" required>
            <br>
            <br>
            <button class="btn btn-primary btn-lg btn-block" style="background-color: #292928;"
type="submit" onclick="validateLogin()">Login</button>
        </div>
    </div>
</div>
<script>
    function validateLogin() {
        var username = document.getElementById('username').value;
        var password = document.getElementById('password').value;
 (username === 'admin' && password === 'adminpass') {
            location.href = 'Home.jsp';
            alert('Welcome, Admin!');
            // Redirect or perform actions for admin login
        } else {
            alert('Invalid credentials. Please try again.');
        }
    }
</script>

    </div>
  </div>
</div>
   <table class="table"  style="background: rgba( 255, 255, 255, 0.5 );
  box-shadow: 0 8px 32px 0 rgba( 31, 38, 135, 0.37 );
  backdrop-filter: blur( 2.5px );
  -webkit-backdrop-filter: blur( 2.5px );
  border-radius: 10px;
  border: 1px solid rgba( 255, 255, 255, 0.18 );">
```

```
      <tbody>
      </tbody>
    </table>
  </div>
    <br>
 <script>
       function submitForm() {
          alert("Great choice! 📚 The book has been added to your cart. Ready to explore more, or proceed to checkout and make it yours?");
       }
    </script>
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js" integrity="sha384-IQsoLXl5PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4GN1R8p" crossorigin="anonymous"></script>
</body>
</html>
```

Registration.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@include file="DatabaseUtils.jsp" %>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>User Registration</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">
</head>
<body style="background-image:url('https://images8.alphacoders.com/476/476792.jpg'); height: 100vh; background-size: cover;">
  <nav class="navbar navbar-expand-lg navbar-dark bg-#AFC8AD">
    <div class="container-fluid pl-5">
       <a class="navbar-brand" href="#"><h1>QuillQuest Book Store</h1></a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
          <li class="nav-item">
            <a class="nav-link active " aria-current="page" href="home2.jsp">Home</a>
```

```
        </li>
         <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="view1.jsp"><b>View
Books</b></a>
        </li>
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="cart.jsp"> Shopping Cart</a>
        </li>
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="payment.jsp">Checkout</a>
        </li>
       </ul>
     </div>
   </div>
  </nav><br>
  <div class="container border py-4 w-50" style="background: rgba( 255, 255, 255, 0.2 );
  box-shadow: 0 8px 32px 0 rgba( 31, 38, 135, 0.37 );
  backdrop-filter: blur( 2.5px );
  -webkit-backdrop-filter: blur( 2.5px );
  border-radius: 10px;
  border: 1px solid rgba( 255, 255, 255, 0.18 );">
    <h6 class="text-center">Ready to write the first chapter of your literary journey? Create your
account now and open the door to a world of books waiting just for you!</h6>
    <form id="registrationForm" method="post" onsubmit="return validateForm()">
     <div class="container mt-1">
      <div class="row justify-content-center">
       <div class="col-md-6">
        <h2 class="mb-4" style="color: #000000;">User Registration</h2>
        <div class="form-group">
         <label for="username" style="color: #000000;"><b>Username:</b></label>
         <input type="text" class="form-control" id="username" name="username" required>
        </div>
        <div class="form-group">
         <label for="email" style="color: #000000;"><b>Email Address:</b></label>
         <input type="email" class="form-control" id="email" name="email" required>
        </div>
        <div class="form-group">
         <label for="password" style="color: #000000;"><b>Password:</b></label>
         <input type="password" class="form-control" id="password" name="password"
required>
        </div>
        <div class="form-group">
         <label for="confirmPassword" style="color:#000000;"><b>Confirm
Password:</b></label>
          <input type="password" class="form-control" id="confirmPassword"
name="confirmPassword" required>
```

```
        </div>
        <br>
        <a type="submit" class="btn btn-primary btn-lg btn-block" style="background-
color:#000000" href="login.jsp">Register</a>
        <br>
        <p class="mb-5 pb-lg-2" style="color: #0000ff;"><b>Already have an account? </b><a
href="login.jsp" style="color: #000000;"><b>Login here</b></a>
       </div>
      </div>
    </div>
  </form>
<%
      if (request.getMethod().equals("POST")) {
         Connection conn = null;
         PreparedStatement pstmt = null;
         try {
            conn = getConnection();
            String username = request.getParameter("username");
            String email = request.getParameter("email");
            String password = request.getParameter("password");
            String confirmpass = request.getParameter("confirmpass");
            String sql = "INSERT INTO users (username, email, password) VALUES (?, ?, ?)";
            pstmt = conn.prepareStatement(sql);
            pstmt.setString(1, username);
            pstmt.setString(2, email);
            pstmt.setString(3, password);
            pstmt.setString(4, confirmpass);

            int result = pstmt.executeUpdate();

            if (result < 1) {
  %>
<script>alert("Invalid id or Character")</script>
<%
            } else {
  %>
<script>alert("Registration Successful")</script>
<%
            }
         } catch (Exception e) {
            out.println("<p>An error occurred: " + e.getMessage() + "</p>");
         } finally {
            try {
               if (pstmt != null) pstmt.close();
               if (conn != null) conn.close();
            } catch (SQLException e) {
               e.printStackTrace();
            }
```

```
      }
    %>
  </div>
  <script>
    function validateForm() {
      var password = document.getElementById("password").value;
      var confirmPassword = document.getElementById("confirmPassword").value;
      if (password !== confirmPassword) {
        alert("Passwords do not match");
        return false;
      }
      return true;
    }
  </script>
  <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
  <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>
```

Login.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@include file="DatabaseUtils.jsp" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Login into QuillQuest Book Store</title>
</head>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
</head>
<body style="background-image:url('https://thumbs.dreamstime.com/b/old-book-flying-letters-
magic-light-background-bookshelf-library-ancient-books-as-symbol-knowledge-history-
218640948.jpg') ; height: 100vh ;background-size: cover;">
  <nav class="navbar navbar-expand-lg navbar-dark bg-#AFC8AD">
    <div class="container-fluid pl-5">
      <a class="navbar-brand" href="#"><h2>QuillQuest Book Store</h2></a>
     <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
       <span class="navbar-toggler-icon"></span>
     </button>
```

```
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="home2.jsp">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="view1.jsp"> View Books</a>
        </li>
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="cart.jsp">Shopping Cart</a>
        </li>
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="payment.jsp">Checkout</a>
        </li>
      </ul>
    </div>
  </div>
 </nav><br>
 <div class="container border py-4 w-50" style="background: rgba( 255, 255, 255, 0.5 );
 box-shadow: 0 8px 32px 0 rgba( 31, 38, 135, 0.37 );
 backdrop-filter: blur( 2.5px );
 -webkit-backdrop-filter: blur( 2.5px );
 border-radius: 10px;
 border: 1px solid rgba( 255, 255, 255, 0.18 );">
  <h3 class="text-center">Login to QuillQuest Bookstore</h3>
  <form method="post"  style="display: grid;place-items: center;">
<body style="background-color:rgb(151, 248, 254);">
                    <div class="container">
   <div class="card" style="border-radius: 1rem;">
      <div class="card-header text-center">Sign Up to get personalized experience &  quick
checkout</div>
      <div class="card-body">
            <div class="form-group">
               <label><b>Email address/Phone no</b></label>
               <input type="email" name="login-email" class="form-control" placeholder="Enter
email">
            </div><br>
            <div class="form-group">
               <label><b>Password</b></label>
               <input type="password" name="login-password" class="form-control"
placeholder="Password">
            </div>
            <div>
               <br>
         <a href="index.jsp" style="text-align: left;">Forgot password?</a>
        </div>
```

```
          <br>
         <!-- Submit button -->
         <button type="submit" class="btn btn-dark">Sign in</button>
          <br>
          <br>
              <p class="mb-5 pb-lg-2" style="color: #393f81;">Don't have an account? <a
href="registration.jsp" style="color: #393f81;">Register here</a>
              <a href="#!" class="small text-muted">Terms of use.</a>
              <a href="#!" class="small text-muted">Privacy policy</a></p>
         </form>
       </div>
   </div>
</div>
   <%
String email = request.getParameter("login-email");
String password = request.getParameter("login-password");
// Check if email and password are provided
if (email != null && password != null) {
   Connection conn = null;
   PreparedStatement pstmt = null;
   ResultSet resultSet = null;
   try {
      conn = getConnection();
      String query = "SELECT * FROM users WHERE email = ? AND password = ?";
      pstmt = conn.prepareStatement(query);
      pstmt.setString(1, email);
      pstmt.setString(2, password);
      resultSet = pstmt.executeQuery();
      if (resultSet.next()) {
         int user_id = resultSet.getInt("user_id");
         HttpSession ss = request.getSession(true);
         ss.setAttribute("user_id", user_id); // User exists, redirect to home page or dashboard
         response.sendRedirect("home2.jsp");
      } else {
         // User does not exist or incorrect credentials, display error message
         out.println("<p>Error: Invalid email/phone or password</p>");
      }
   } catch (SQLException e) {
      e.printStackTrace();
   } finally {
      // Close database resources
      try {
         if (resultSet != null) {
            resultSet.close();
         }
         if (pstmt != null) {
```

```
            pstmt.close();
        }
        if (conn != null) {
            conn.close();
        }
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
  }
}
%>
    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js"
integrity="sha384-
IQsoLXl5PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4GN1R8p"
crossorigin="anonymous"></script>
    <script>
    function submitForm() {
        // You can add your payment processing logic here
        alert("Great to see you again! ✨ Your account is now active, and your bookshelf is
ready for new additions. Enjoy the reading journey!");
    }
  </script>
    </body>
</html>
```

Payment.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@include file="DatabaseUtils.jsp" %>
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Online Store Payment</title>
  <link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
</head>

<body style="background-image:url('https://images4.alphacoders.com/132/1326368.png') ;
height: 100vh ;background-size: cover;">
 <nav class="navbar navbar-expand-lg navbar-dark bg-#AFC8AD">
  <div class="container-fluid pl-5">
    <a class="navbar-brand" href="#"><h1>QuillQuest Book Store</h1></a>
```

```
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
     <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
      <li class="nav-item">
       <a class="nav-link active " aria-current="page" href="home2.jsp">Home</a>
      </li>
       <li class="nav-item">
        <a class="nav-link active" aria-current="page" href="view1.jsp"><b>Available
Books</b></a>
      </li>
      <li class="nav-item">
       <a class="nav-link active" aria-current="page" href="cart.jsp"> Shopping Cart</a>
      </li>
      <li class="nav-item">
       <a class="nav-link active" aria-current="page" href="index.jsp">Logout</a>
      </li>
     </ul>
    </div>
  </div>
 </nav><br>

 <div class="container border py-4 w-50" style="background: rgba( 255, 255, 255, 0.3 );
 box-shadow: 0 8px 32px 0 rgba( 31, 38, 135, 0.37 );
 backdrop-filter: blur( 2.5px );
 -webkit-backdrop-filter: blur( 2.5px );
 border-radius: 10px;
 border: 1px solid rgba( 255, 255, 255, 0.18 );">
  <h4 class="text-center" style="color: #000000;">Don't leave your book hanging! Seal the deal
on your literary treasure with a smooth checkout. Happy reading awaits</h4>
  <form action="user-login" method="post"  style="display: grid;place-items: center;">

  <div class="container mt-2">
    <div class="row justify-content-center">
      <div class="col-md-8">
        <form id="paymentForm">
          <h2 class="mb-4" style="color: #000000;">Payment Details</h2>
          <div class="form-row">
            <div class="form-group col-md-6">
              <label for="firstName" style="color: #000000;"><b>First Name:</b></label>
              <input type="text" class="form-control" id="firstName" name="firstName"
required>
            </div>
            <div class="form-group col-md-6">
              <label for="lastName" style="color: #000000;"><b>Last Name:</b></label>
```

```
                    <input type="text" class="form-control" id="lastName" name="lastName"
required>
                </div>
            </div>
            <div class="form-group">
                <label for="email" style="color: #000000;"><b>Email Address:</b></label>
                <input type="email" class="form-control" id="email" name="email" required>
            </div>
            <div class="form-group">
                <label for="address" style="color: #000000;"><b>Billing Address:</b></label>
                <input type="text" class="form-control" id="address" name="address" required>
            </div>
            <div class="form-row">
                <div class="form-group col-md-6">
                    <label for="cardNumber" style="color: #000000;"><b>Card
Number:</b></label>
                    <input type="text" class="form-control" id="cardNumber"
name="cardNumber" placeholder="1234 5678 9012 3456" required>
                </div>
                <div class="form-group col-md-6">
                    <label for="expiryDate" style="color: #000000;"><b>Expiry
Date:</b></label>
                    <input type="date" class="form-control" id="expiryDate" name="expiryDate"
placeholder="MM/YY" required>
                </div>
            </div>
            <div class="form-group">
                <label for="cvv" style="color: #000000;"><b>CVV:</b></label>
                <input type="text" class="form-control" id="cvv" name="cvv"
placeholder="123" required>
            </div>
            <a class="btn btn-success" onclick="submitForm()" href="home2.jsp">Complete
Purchase</a>
        </form>
    </div>
</div>
</div>
<br>
<%
if (request.getMethod().equals("POST")) {
    Connection conn = null;
    PreparedStatement pstmt = null;
    try {
        conn = getConnection(); // Get connection from DatabaseUtils.jsp
        String firstname = request.getParameter("firstName"); // corrected parameter name
        String lastname = request.getParameter("lastName"); // corrected parameter name
```

```
        String email = request.getParameter("email");
        String billaddress = request.getParameter("address"); // corrected parameter name
        String cardNumber = request.getParameter("cardNumber"); // corrected parameter name
        String expiryDate = request.getParameter("expiryDate"); // corrected parameter name
        String cvv = request.getParameter("cvv");
        String sql = "INSERT INTO payment (firstname, lastname, email, billaddress, cardno,
expiredate, cvv) VALUES (?, ?, ?, ?, ?, ?, ?)";
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, firstname);
        pstmt.setString(2, lastname);
        pstmt.setString(3, email);
        pstmt.setString(4, billaddress);
        pstmt.setString(5, cardNumber);
        pstmt.setString(6, expiryDate); // corrected index
        pstmt.setString(7, cvv); // corrected index
      int result = pstmt.executeUpdate();
        if (result < 1) {
%>
<script>alert("Insertion Failed")</script>
<%
        } else {
%>
<script>alert("Insertion Successful")</script>
<%
        }
    } catch (Exception e) {
        out.println("<p>An error occurred: " + e.getMessage() + "</p>");
    } finally {
        try {
            if (pstmt != null) pstmt.close();
            if (conn != null) conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
}
%>
    <!-- Bootstrap JS and Popper.js -->
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
        function submitForm() {
            alert("Thank you for choosing us to be your bookish companion! 📚✨📖 Your order is
confirmed, and your books are gearing up to embark on a journey to your doorstep. Happy
reading!");
        }
    </script>
</body>
</html>
```

View.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@include file="DatabaseUtils.jsp" %>
<!DOCTYPE html>
<html>
<head>
   <meta charset="UTF-8">
   <title>View Data</title>
   <style>
   </style>
   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
</head>
<body style="background-image:url('https://images4.alphacoders.com/132/1326368.png') ;
height: 100vh ;background-size: cover;">
  <nav class="navbar navbar-expand-lg navbar-dark bg-#AFC8AD">
   <div class="container-fluid pl-5">
      <a class="navbar-brand" href="#"><h1>QuillQuest Book Store</h1></a>
     <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
       <span class="navbar-toggler-icon"></span>
     </button>
     <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="Home.jsp">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="insert.jsp">Insert Books</a>
        </li>
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="update.jsp"> Update Books</a>
        </li>
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="delete.jsp"> Delete Books</a>
        </li>
         <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="index.jsp">Logout</a>
        </li>
      </ul>
     </div>
   </div>
  </nav><br><br>
```

```
<div class="container border py-4 w-50" style="background: rgba( 255, 255, 255, 0.2);
box-shadow: 0 8px 32px 0 rgba( 31, 38, 135, 0.37 );
backdrop-filter: blur( 2.5px );
-webkit-backdrop-filter: blur( 2.5px );
border-radius: 10px;
border: 1px solid rgba( 255, 255, 255, 0.18 );">
  <h3 class="text-center" style="background:#e8ebe9">View Books</h3>
  <table class="table table-hover" style="background:#e8ebe9">
    <thead>
      <tr>
        <th>ID</th>
        <th>Book Name</th>
        <th>Price</th>
        <th>Author/Publisher</th>
      </tr>
    </thead>
    <tbody>
      <%
        Connection conn = null;
        PreparedStatement pstmt = null;
        ResultSet rs = null;
        try {
          conn = getConnection();
          String sql = "SELECT * FROM books";
          pstmt = conn.prepareStatement(sql);
          rs = pstmt.executeQuery();
          while (rs.next()) {
            int book_id = rs.getInt("book_id");
            String bookname = rs.getString("bookname");
            int price = rs.getInt("price");
            String author = rs.getString("author");
            out.println("<tr>");
            out.println("<td>" + book_id + "</td>");
            out.println("<td>" + bookname + "</td>");
            out.println("<td>" + price + "</td>");
            out.println("<td>" + author + "</td>");
            out.println("</tr>");
          }
        } catch (Exception e) {
          e.printStackTrace();
        } finally {
          try {
            if (rs != null) rs.close();
            if (pstmt != null) pstmt.close();
            if (conn != null) conn.close();
          } catch (SQLException e) {
```

```
                e.printStackTrace();
            }
        }
    %>
    </tbody>
  </table>
 </div>
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js"
integrity="sha384-
cVKIPhGWiC2Al4u+LWgxfKTRIcfu0JTxR+EQDz/bgldoEyl4H0zUF0QKbrJ0EcQF"
crossorigin="anonymous"></script>
</body>
</html>
```

Insert.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@include file="DatabaseUtils.jsp" %>
<jsp:directive.page
import="java.io.*,java.sql.*,javax.servlet.annotation.*,javax.servlet.http.*,javax.servlet.*" />
<jsp:directive.page
import="java.io.*,java.sql.*,javax.servlet.annotation.*,javax.servlet.http.*,javax.servlet.*,javax.s
ervlet.annotation.MultipartConfig" />

<!DOCTYPE html>
<html>
<head>
   <meta charset="UTF-8">
   <title>Insert Books</title>
   <style>

   </style>
   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
</head>
<body style="background-image:url('https://wallpapers.com/images/hd/education-open-book-
blackboard-24neqyjtkzc2rc21.jpg') ; height: 100vh ;background-size: cover;">
   <nav class="navbar navbar-expand-lg navbar-dark bg-#AFC8AD">
      <div class="container-fluid pl-5">
         <a class="navbar-brand" href="#"><h1>QuillQuest Online Book Store</h1></a>
         <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
         </button>
         <div class="collapse navbar-collapse" id="navbarSupportedContent">
```

```
            <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
               <li class="nav-item">
                  <a class="nav-link active " aria-current="page" href="Home.jsp">Home</a>
               </li>
               <li class="nav-item">
                  <a class="nav-link active" aria-current="page" href="view.jsp"> View Books</a>
               </li>
               <li class="nav-item">
                  <a class="nav-link active" aria-current="page" href="update.jsp">Update
Books</a>
               </li>
               <li class="nav-item">
                  <a class="nav-link active " aria-current="page" href="delete.jsp">Delete
Books</a>
               </li>
               <li class="nav-item">
                  <a class="nav-link active" aria-current="page"
href="index.jsp"><b>Logout</b></a>
               </li>
            </ul>
         </div>
      </div>
   </nav><br>
   <br>
   <div class="container border py-4 w-50" style="background: rgba( 255, 255, 255, 0.5 );
   box-shadow: 0 8px 32px 0 rgba( 31, 38, 135, 0.37 );
   backdrop-filter: blur( 2.5px );
   -webkit-backdrop-filter: blur( 2.5px );
   border-radius: 10px;
   border: 1px solid rgba( 255, 255, 255, 0.18 );">
      <h3 class="text-center">Insert Books</h3>
      <form method="post" action="" style="display: grid;place-items: center;">
         <div class="mb-3 w-75">
            <label for="exampleInputText1" class="form-label"><b>Book Id:</b> </label>
            <input type="text" class="form-control" name="book_id">
         </div>
         <div class="mb-3 w-75">
            <label for="exampleInputText1" class="form-label"><b>Book Name: </b></label>
            <input type="text" class="form-control" name="bookname">
         </div>
         <div class="mb-3 w-75">
            <label for="exampleInputText1" class="form-label"><b>Author/Publisher:
</b></label>
            <input type="text" class="form-control" name="author">
         </div>
         <div class="mb-3 w-75">
```

```
            <label for="exampleInputText1" class="form-label"><b>Price:  </b></label>
            <input type="text" class="form-control" name="price">
        </div>
        <div class="mb-3 w-75">
            <label for="exampleInputText1" class="form-label"><b>Image of Book:
</b></label>
            <input type="file" class="form-control" name="bookImage" id="bookimage">
            <input type="text" hidden class="form-control" name="image" id="image">
        </div>
        <button type="submit" class="btn btn-primary" style="width: 25%;"
onclick="ok()">Insert</button>
    </form>
  </div>
  <br>
  <%
    if (request.getMethod().equals("POST")) {
        Connection conn = null;
        PreparedStatement pstmt = null;
        try {
            conn = getConnection(); // Get connection from DatabaseUtils.jsp
            int book_id = Integer.parseInt(request.getParameter("book_id"));
            String bookname = request.getParameter("bookname");
            String author = request.getParameter("author");
             String price = request.getParameter("price");
            String image = request.getParameter("image");
            String sql = "INSERT INTO books (id, bookname, author, price,image) VALUES (?,
?, ?, ?,?)";
            pstmt = conn.prepareStatement(sql);
            pstmt.setInt(1, book_id);
            pstmt.setString(2, bookname);
            pstmt.setString(3, author);
            pstmt.setString(4, price);
            pstmt.setString(5, image);
            int result = pstmt.executeUpdate();
            if (result < 1) {
  %>
<script>alert("Invalid id or Character")</script>
<%
            } else {
  %>
<script>alert("Insertion Successful")</script>
<%
            }
        } catch (Exception e) {
            out.println("<p>An error occurred: " + e.getMessage() + "</p>");
        } finally {
```

```
            try {
                if (pstmt != null) pstmt.close();
                if (conn != null) conn.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
%>
<script>
    function ok() {
var image = document.getElementById('image');
var bookimage = document.getElementById('bookimage').value;
var imageName = bookimage.split('\\').pop(); // Extract image name from the URL
image.value = imageName; // Set the image source to the extracted image name
}
</script>
</body>
</html>
```

Update.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@include file="DatabaseUtils.jsp" %>

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Update Books Here</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
</head>
<body style="background-image:url('https://trumpwallpapers.com/wp-content/uploads/Book-
Wallpaper-03-3840-x-2400.jpg') ; height: 100vh ;background-size: cover;">
<nav class="navbar navbar-expand-lg navbar-dark bg-#88AB8E ">
    <div class="container-fluid pl-5">
        <a class="navbar-brand" href="#"><h1>QuillQuest Online Book Store</h1></a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarSupportedContent">
            <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
```

```
            <li class="nav-item">
              <a class="nav-link active" aria-current="page" href="Home.jsp"> Home</a>
            </li>
            <li class="nav-item">
              <a class="nav-link active" aria-current="page" href="insert.jsp">Insert Books</a>
            </li>
            <li class="nav-item">
              <a class="nav-link active" aria-current="page" href="view.jsp">View Books</a>
            </li>
            <li class="nav-item">
              <a class="nav-link active" aria-current="page" href="delete.jsp">Delete Books</a>
            </li>
            <li class="nav-item">
              <a class="nav-link active" aria-current="page" href="index.jsp">Logout</a>
            </li>
          </ul>
        </div>
    </div>
</nav><br><br>
<div class="container border py-4 w-50" style="background: rgba( 255, 255, 255, 0.5 );
box-shadow: 0 8px 32px 0 rgba( 31, 38, 135, 0.37 );
backdrop-filter: blur( 2.5px );
-webkit-backdrop-filter: blur( 2.5px );
border-radius: 10px;
border: 1px solid rgba( 255, 255, 255, 0.18 );">
<h3 class="text-center">Update Books Here....</h3>
<form method="post" style="display: grid;place-items: center;">
    <div class="mb-3 w-75">
        <label for="exampleInputText1" class="form-label"><b>Existing Book Id: </b></label>
        <input type="text" class="form-control" name="id">
    </div>
    <div class="mb-3 w-75">
        <label for="exampleInputText1" class="form-label"><b>New Book Name: </b></label>
        <input type="text" class="form-control" name="newbookname">
    </div>
    <div class="mb-3 w-75">
        <label for="exampleInputText1" class="form-label"><b>New Price: </b></label>
        <input type="text" class="form-control" name="newprice">
    </div>
    <div class="mb-3 w-75">
        <label for="exampleInputText1" class="form-label"><b>New Author/Publisher:
</b></label>
        <input type="text" class="form-control" name="newauthor">
    </div>
    <button type="submit" class="btn btn-primary" style="width: 25%;">Update</button>
</form>
```

```
</div>
<br>
<%
if (request.getMethod().equals("POST")) {
    Connection conn = null;
    PreparedStatement pstmt = null;
    try {
        String book_id = request.getParameter("book_id");
        String newbookname = request.getParameter("newbookname");
        int newprice = Integer.parseInt(request.getParameter("newprice"));
        String newauthor = request.getParameter("newauthor");

        conn = getConnection();
        String sql = "UPDATE books SET bookname=?, price=?, author=? WHERE id=?";
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, newbookname);
        pstmt.setInt(2, newprice);
        pstmt.setString(3, newauthor);
        pstmt.setString(4, book_id);

        int rowsUpdated = pstmt.executeUpdate();

        if (rowsUpdated > 0) {
%><script>alert("Update Successful")</script><%
        } else {
%><script>alert("Update Failed")</script><%
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        try {
            if (pstmt != null) pstmt.close();
            if (conn != null) conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
}
%>
</body>
</html>
```

Delete.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@include file="DatabaseUtils.jsp" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
```

```
    <title>Delete Books</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
</head>
<body style="background-image:url('https://images5.alphacoders.com/133/1338186.png') ;
height: 100vh ;background-size: cover;">
  <nav class="navbar navbar-expand-lg navbar-dark bg-#AFC8AD">
    <div class="container-fluid pl-5">
       <a class="navbar-brand" href="#"><h1>QuillQuest Book Store</h1></a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
          <li class="nav-item">
           <a class="nav-link active " aria-current="page" href="Home.jsp">Home</a>
          </li>
           <li class="nav-item">
           <a class="nav-link active" aria-current="page" href="insert.jsp">Insert Books</a>
          </li>
          <li class="nav-item">
           <a class="nav-link active" aria-current="page" href="view.jsp">View Books</a>
          </li>
          <li class="nav-item">
           <a class="nav-link active" aria-current="page" href="update.jsp">Update Books</a>
          </li>
         <li class="nav-item">
           <a class="nav-link active" aria-current="page" href="index.jsp">Logout</a>
          </li>
        </ul>
      </div>
    </div>
  </nav><br>
<br>
<br>
  <div class="container border py-4 w-50" style="background: rgba( 255, 255, 255, 0.5 );
  box-shadow: 0 8px 32px 0 rgba( 31, 38, 135, 0.37 );
  backdrop-filter: blur( 2.5px );
```

```
  -webkit-backdrop-filter: blur( 2.5px );
  border-radius: 10px;
  border: 1px solid rgba( 255, 255, 255, 0.18 );">
    <h3 class="text-center">Delete Books</h3>
    <form  style="display: grid;place-items: center;" method="post">
      <div class="mb-3 w-75">
        <div class="mb-3">
            <label for="exampleInputText1" class="form-label"><b>Book Id For Deletion:
</b></label>
            <input type="text" class="form-control" name="id">
        </div>
    </div>
        <button type="submit" class="btn btn-primary"  style="width: 25%;">Delete</button>
      </form>
  </div>
    <br>
    <%
    if (request.getMethod().equals("POST")) {
        Connection conn = null;
        PreparedStatement pstmt = null;
        try {
            int book_id = Integer.parseInt(request.getParameter("book_id"));
            conn = getConnection();
            String sql = "delete from books where id=?";
            pstmt = conn.prepareStatement(sql);
            pstmt.setInt(1, book_id);
            int deleteResult = pstmt.executeUpdate(); // Use executeUpdate() for DELETE statement
            if (deleteResult > 0) {
                %><script>alert("Delete Successful")</script><%
            } else {
                %><script>alert("No row deleted. Make sure this book exists.")</script><%
            }
        } catch (Exception e) {
            out.println("<p>An error occurred: " + e.getMessage() + "</p>");
        } finally {
            try {
                if (pstmt != null) pstmt.close();
                if (conn != null) conn.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
%>
</body>
</html>
```

ContactUs.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@include file="DatabaseUtils.jsp" %>
<jsp:directive.page
import="java.io.*,java.sql.*,javax.servlet.annotation.*,javax.servlet.http.*,javax.servlet.*" />
<jsp:directive.page
import="java.io.*,java.sql.*,javax.servlet.annotation.*,javax.servlet.http.*,javax.servlet.*,javax.s
ervlet.annotation.MultipartConfig" />
<!DOCTYPE html>
<html>
<head>
   <meta charset="UTF-8">
   <title>Contact Us</title>
   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
</head>
<body style="background-image:url('https://www.shutterstock.com/image-photo/website-page-
contact-us-email-600nw-1958537320.jpg') ; height: 100vh ;background-size: cover;">
   <nav class="navbar navbar-expand-lg navbar-dark bg-#AFC8AD">
      <div class="container-fluid pl-5">
         <a class="navbar-brand" href="#"><h1>QuillQuest Online Book Store</h1></a>
         <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
         </button>
         <div class="collapse navbar-collapse" id="navbarSupportedContent">
            <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
               <li class="nav-item">
         <a class="nav-link active " aria-current="page" href="home2.jsp">Home</a>
        </li>
         <li class="nav-item">
         <a class="nav-link active" aria-current="page" href="view1.jsp"><b>Available
Books</b></a>
        </li>
        <li class="nav-item">
         <a class="nav-link active" aria-current="page" href="cart.jsp"> Shopping Cart</a>
        </li>
        <li class="nav-item">
         <a class="nav-link active" aria-current="page" href="index.jsp">Logout</a>
        </li>
            </ul>
         </div>
      </div>
```

```
</nav><br>
<div class="container border py-3  w-50" style="background: rgba( 255, 255, 255, 0.5 );
box-shadow: 0 8px 32px 0 rgba( 31, 38, 135, 0.37 );
backdrop-filter: blur( 2.5px );
-webkit-backdrop-filter: blur( 2.5px );
border-radius: 10px;
border: 1px solid rgba( 255, 255, 255, 0.18 );">
<div class="container mt-0">
  <div class="row justify-content-center">
    <div class="col-md-10">
      <section class="mb-1">
        <h2 class="h1-responsive font-weight-bold text-center my-2">Contact us</h2>
        <p class="text-center w-responsive mx-auto mb-3">Do you have any questions?
Please do not hesitate to contact us directly. Our team will come back to you within a matter of
hours to help you.</p>
        <div class="row">
          <div class="col-md-5 col-md-10 mb-md-1 mb-3 ">
              <h6>Our Contact Details </h6>
                <p>Address: Navi Mumbai, Maharashtra 410206, India</b></p>
                <p>Phone No: +9123456789</p>
                <p>Email: pranjalshinde1219@gmail.com</p>
          </div>
          <h5>You can ask your Questions By filling following form</h5>
          <div class="col-md-9 mb-md-0 mb-5">
            <form id="contact-form" name="contact-form" method="POST">
              <div class="row">
                <div class="col-md-6">
                  <div class="md-form mb-0">
                    <label for="name" class="">Your name</label>
                    <input type="text" id="name" name="name" class="form-control">
                  </div>
                </div>
                <div class="col-md-6">
                  <div class="md-form mb-0">
                    <label for="email" class="">Your email</label>
                     <input type="text" id="email" name="email" class="form-control">
                  </div>
                </div>
              </div>
              <div class="row">
                <div class="col-md-12">
                  <div class="md-form mb-0">
                    <label for="subject" class="">Subject</label>
                    <input type="text" id="subject" name="subject" class="form-control">
                  </div>
                </div>
```

```
                    </div>
                    <div class="row">
                       <div class="col-md-12">
                          <div class="md-form">
                             <label for="message">Your message</label>
                              <textarea type="text" id="message" name="message" rows="2"
class="form-control md-textarea"></textarea>
                          </div>
                       </div>
                    </div>
                    <br>
                    <div class="text-center text-md-left">
                       <button type="submit" class="btn btn-primary">Send</button>
                    </div>
                 </form>
                 <br>
                 <div class="status"></div>
              </div>
           </div>
        </section>
     </div>
  </div>
</div>
  <%
     if (request.getMethod().equals("POST")) {
        Connection conn = null;
        PreparedStatement pstmt = null;
        try {
           conn = getConnection(); // Get connection from DatabaseUtils.jsp
           String name = request.getParameter("name");
           String email = request.getParameter("email");
            String subject = request.getParameter("subject");
           String message = request.getParameter("message");
           String sql = "INSERT INTO contactus (name, email, subject, message) VALUES (?, ?,
?, ?)";
           pstmt = conn.prepareStatement(sql);
           pstmt.setString(1, name);
           pstmt.setString(2, email);
           pstmt.setString(3, subject);
           pstmt.setString(4, message);
           int result = pstmt.executeUpdate();
           if (result < 1) {
  %>
<script>alert("Invalid id or Character")</script>
<%
           } else {
```

```
   %>
<script>alert("Insertion Successful")</script>
<%
            }
        } catch (Exception e) {
            out.println("<p>An error occurred: " + e.getMessage() + "</p>");
        } finally {
            try {
                if (pstmt != null) pstmt.close();
                if (conn != null) conn.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
   %>
   <script>
      function ok() {
   var image = document.getElementById('images');
   var bookimage = document.getElementById('bookimage').value;
   var imageName = bookimage.split('\\').pop(); // Extract image name from the URL
   image.value = imageName; // Set the image source to the extracted image name
}
   </script>
</body>
</html>
```

Feedback.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@include file="DatabaseUtils.jsp" %>
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Online Store Payment</title>
   <link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body style="background-image:url('https://images4.alphacoders.com/132/1326368.png') ;
height: 100vh ;background-size: cover;">
  <nav class="navbar navbar-expand-lg navbar-dark bg-#AFC8AD">
   <div class="container-fluid pl-5">
      <a class="navbar-brand" href="#"><h1>QuillQuest Book Store</h1></a>
```

```
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
     <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
       <li class="nav-item">
         <a class="nav-link active " aria-current="page" href="home2.jsp">Home</a>
       </li>
        <li class="nav-item">
         <a class="nav-link active" aria-current="page" href="view1.jsp"><b>Available
Books</b></a>
       </li>
       <li class="nav-item">
         <a class="nav-link active" aria-current="page" href="cart.jsp"> Shopping Cart</a>
       </li>
       <li class="nav-item">
             <a class="nav-link active" aria-current="page" href="contactUs.jsp"><b>Contact
Us</b></a>
           </li>
       <li class="nav-item">
         <a class="nav-link active" aria-current="page" href="login.jsp">Login</a>
       </li>
     </ul>
    </div>
  </div>
 </nav><br>
 <div class="container border py-4 w-50" style="background: rgba( 255, 255, 255, 0.3 );
 box-shadow: 0 8px 32px 0 rgba( 31, 38, 135, 0.37 );
 backdrop-filter: blur( 2.5px );
 -webkit-backdrop-filter: blur( 2.5px );
 border-radius: 10px;
 border: 1px solid rgba( 255, 255, 255, 0.18 );">
  <h4 class="text-center" style="color: #000000;">Your feedback is a gift. It helps us improve
and serve you better.</h4>
  <form method="post"  style="display: grid;place-items: center;">
  <div class="container mt-2">
   <div class="row justify-content-center">
     <div class="col-md-8">
       <form id="feedbackForm">
         <h2 class="mb-4" style="color: #000000;">Leave Feedback</h2>
         <div class="form-row">
           <div class="form-group col-md-6">
             <label for="firstName" style="color: #000000;"><b>First Name:</b></label>
```

```
                <input type="text" class="form-control" id="firstname" name="firstname"
required>
            </div>
            <div class="form-group col-md-6">
                <label for="lastName" style="color: #000000;"><b>Last Name:</b></label>
                <input type="text" class="form-control" id="lastname" name="lastname"
required>
            </div>
        </div>
        <div class="form-group">
            <label for="email" style="color: #000000;"><b>Email Address:</b></label>
            <input type="email" class="form-control" id="email" name="email" required>
        </div>
        <div class="form-group">
            <label for="feedback" style="color: #000000;"><b>Feedback:</b></label>
            <textarea class="form-control" id="feedback" name="feedback" rows="5"
required></textarea>
        </div>
        <button type="submit" class="btn btn-success" >Submit Feedback</button>
    </form>
  </div>
</div>
<%
if (request.getMethod().equals("POST")) {
    Connection conn = null;
    PreparedStatement pstmt = null;
    try {
        conn = getConnection(); // Get connection from DatabaseUtils.jsp
        String firstname = request.getParameter("firstname"); // corrected parameter name
        String lastname = request.getParameter("lastname"); // corrected parameter name
        String email = request.getParameter("email");
        String feedback = request.getParameter("feedback");
        String sql = "INSERT INTO feedback (firstname, lastname, email, feedback) VALUES (?,
?, ?, ?)";
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, firstname);
        pstmt.setString(2, lastname);
        pstmt.setString(3, email);
        pstmt.setString(4, feedback);
        int result = pstmt.executeUpdate();
        if (result < 1) {
%>
<script>alert("Feedback Failed")</script>
<%
        } else {
%>
```

```
<script>alert("Feedback Successful")</script>
<%
        }
    } catch (SQLException e) {
        e.printStackTrace(); // Or log the exception
    } finally {
        try { if (pstmt != null) pstmt.close(); } catch (Exception e) { }
        try { if (conn != null) conn.close(); } catch (Exception e) { }
    }
}
%>
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
        function submitForm()
            alert("Thank you for sharing your feedback with us! Your insights are invaluable and help us enhance your experience at our bookstore. We truly appreciate your time and effort in helping us serve you better.");
        }
    </script>
</body>
</html>
```

Cart.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@include file="DatabaseUtils.jsp" %>
<%@ page import="java.util.ArrayList" %>
<%@ page import="java.util.Map" %>
<%@ page import="java.util.Map.Entry" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Shopping Cart</title>
</head>
<body style="background-image:url('https://images4.alphacoders.com/132/1326368.png') ; height: 100vh ;background-size: cover;">
<nav class="navbar navbar-expand-lg navbar-dark bg-#AFC8AD">
    <div class="container-fluid pl-5">
        <a class="navbar-brand" href="#"><h1>QuillQuest Book Store</h1></a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarSupportedContent">
            <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
```

```
            <li class="nav-item">
               <a class="nav-link active " aria-current="page" href="home2.jsp">Home</a>
            </li>
            <li class="nav-item">
               <a class="nav-link active" aria-current="page" href="view1.jsp"> View Books</a>
            </li>
             <li class="nav-item">
               <a class="nav-link active" aria-current="page"
href="payment.jsp"><b>Payment</b></a>
            </li>
            <li class="nav-item">
               <a class="nav-link active" aria-current="page" href="contactUs.jsp"><b>Contact
Us</b></a>
            </li>
            <li class="nav-item">
               <a class="nav-link active" aria-current="page" href="index.jsp">Logout</a>
            </li>
      </div>
   </div>
</nav><br>
<div class="container mt-2">
   <h2 class="mb-3" style="color: #ddd;">Shopping Cart</h2>
   <div class="row">
      <div class="col-md-8">
         <%
            java.util.Map<String, Integer> cart = (java.util.Map<String, Integer>)
session.getAttribute("cart");
            if (cart != null && !cart.isEmpty()) {
               for (java.util.Map.Entry<String, Integer> entry : cart.entrySet()) {
         %>
         <div class="card mb-3 product-item">
            <div class="card-body">
               <h5 class="card-title"> <%= entry.getKey() %></h5>
               <p class="card-text price-per-unit">Price: Rs. <%= entry.getValue() %></p>
               <label for="quantity">Quantity:</label>
               <div style="display:flex;flex-direction:row;align-items:center;justify-content:space-
between;width: 100%;">
                  <button class="btn btn-primary btn-sm quantity-increase"
style="padding:12px;">+</button>
                  <input type="number" id="quantity" class="form-control mb-2 quantity-input"
style="width:90%" value="1" min="1">
                  <button class="btn btn-danger btn-sm quantity-decrease"
style="padding:12px;">-</button>
               </div>
            </div>
         <%
```

```
                }
            } else {
        %>
        <div class="alert alert-info" role="alert">
            Your cart is empty. <a href="home2.jsp" class="alert-link">Browse books</a> to add
items to your cart.
        </div>
        <div class="col-md-12">
            <div class="card">
                <div class="card-body">
                    <h5 class="card-title">Order Summary</h5>
                    <ul class="list-group">
                        <!-- Additional products can be added similarly -->
                        <li class="list-group-item d-flex justify-content-between align-items-center">
                            Total
                            <span id="total-price" class="badge badge-success badge-pill text-dark">Rs.
<%= calculateTotal(cart) %></span>
                        </li>
                    </ul>
                    <a class="btn btn-success btn-block mt-3 chkout" >Proceed to Checkout</a>
                </div>
        </div>
</div>
<br><br>
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script>
    var tp=0;
    var chk=document.querySelector('.chkout');
    function calculateTotalPrice() {
        var totalPrice = 0;
        $(".product-item").each(function() {
            var quantity = parseInt($(this).find(".quantity-input").val());
            var pricePerUnit = parseFloat($(this).find(".price-per-unit").text().replace("Price: Rs. ",
""));
            var totalPricePerItem = quantity * pricePerUnit;
            totalPrice += totalPricePerItem;
        });
        $("#total-price").text("Rs. " + totalPrice.toFixed(2));
        tp=totalPrice
    }
    $(".quantity-increase").click(function() {
        var inputField = $(this).siblings(".quantity-input");
        var currentValue = parseInt(inputField.val());
        inputField.val(currentValue + 1);
        calculateTotalPrice();
    });
```

```
    $(".quantity-decrease").click(function() {
        var inputField = $(this).siblings(".quantity-input");
        var currentValue = parseInt(inputField.val());
        if (currentValue > 1) {
            inputField.val(currentValue - 1);
            calculateTotalPrice();
        }
    });
    $(".quantity-input").change(function() {
        calculateTotalPrice();
    });
    $(document).ready(function() {
        calculateTotalPrice();
    });
    chk.addEventListener('click',()=>{
        window.location.href='checkout.jsp?totalPrice='+tp;
    })
</script>
</body>
</html>
<%!
    private int calculateTotal(java.util.Map<String, Integer> cart) {
        int total = 0;
        if (cart != null && !cart.isEmpty()) {
            for (java.util.Map.Entry<String, Integer> entry : cart.entrySet()) {
                total += entry.getValue();
            }
        return total;
    }
%>
```

AddToCart.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    String product = request.getParameter("product");
    int price = Integer.parseInt(request.getParameter("price"));
    java.util.Map<String, Integer> cart = (java.util.Map<String, Integer>)
session.getAttribute("cart");
    if (cart == null) {
        cart = new java.util.HashMap<String, Integer>();
        session.setAttribute("cart", cart);
    }
    cart.put(product, price);
    response.sendRedirect("cart.jsp");
%>
```

7.3 VALIDATIONS

For this project, we will typically need various validations to ensure data integrity, security, and a smooth user experience. Here are some key validations we might consider implementing:

Form Field Validations:
Ensure all required fields are filled out.
Validate input formats (e.g., email format, phone number format).
Check for input length limits and data types (e.g., numeric fields should only accept numbers).
Implement client-side and server-side validations to prevent malicious input.

Authentication and Authorization:
Validate user credentials during login.
Implement session management to restrict access to certain pages based on user roles.
Protect sensitive operations with proper authorization checks.

Data Validations:
Validate data integrity before inserting or updating records in the database.
Implement referential integrity constraints to maintain data consistency.
Check for duplicate records to avoid data redundancy.
File Upload Validations:
Validate file types and sizes for uploaded images or documents.
Implement server-side checks to prevent uploading of potentially harmful files.

Cart Validations:
Ensure items added to the cart are available in the inventory.
Validate quantities to prevent adding more items than available in stock.
Implement checks to handle concurrent cart modifications by multiple users.

Payment Validations:
Validate payment information (e.g., credit card numbers, expiration dates) before processing transactions.
Implement secure payment gateways to prevent fraudulent transactions.
Verify billing and shipping details for accuracy.

Error Handling:
Implement error handling mechanisms to display user-friendly error messages.
Log errors for debugging and auditing purposes.

Cross-Site Scripting (XSS) and Cross-Site Request Forgery (CSRF) Prevention:
Sanitize user input to prevent XSS attacks.
Implement CSRF tokens to prevent CSRF attacks.

Data Encryption:
Encrypt sensitive data (e.g., user passwords, payment information) to protect it from unauthorized access.
Use secure communication protocols (HTTPS) to transmit sensitive data over the network.

By implementing these validations, we can enhance the security, reliability, and usability of our online bookstore website, providing users with a safer and more enjoyable shopping experience.

CHAPTER – 8 (TESTING)

8.1 TEST CASES

Test Scenario ID: 1
Tester Name: Pranjal Shinde
Test Page: Admin login page of Online Bookstore website

Test ID	Scenario	Test Case steps	Test Data	Actual Output	Expected Output	Status
1	Enter the correct admin username	1.Open Website	Login to QuillQuest			
	and password	2. Open admin.jsp page	http://localhost:8080/ppranjal/admin.jsp	Redirect to the index.jsp	Redirect to the index.jsp	PASS
		3.Enter valid username	admin	page	page	
		4. Enter correct password	adminpass			
		5. click on login				
2	Enter correct username and	1.Open Website	Login to QuillQuest			
	incorrect password	2. Open admin.jsp page	http://localhost:8080/ppranjal/admin.jsp	Invalid credentials message	Invalid credentials message	PASS
		3.Enter valid username	admin	will pop up at top of the page	will pop up at top of the page	
		4. Enter incorrect password	12345			
		5. click on login				
3	Enter incorrect username and	1.Open Website	Login to QuillQuest	Invalid credentials message	Invalid credentials message	PASS
	correct password	2. Open admin.jsp page	http://localhost:8080/ppranjal/admin.jsp	will pop up at top of the page	will pop up at top of the page	
		3.Enter incorrect username	pranjal			
		4. Enter correct password	adminpass			
		5. click on login				
4	Enter empty username and	1.Open Website	Login to QuillQuest	Invalid credentials message	Invalid credentials message	PASS
	password both	2. Open admin.jsp page	http://localhost:8080/ppranjal/admin.jsp	will pop up at top of the page	will pop up at top of the page	
		3.Empty username				
		4. Empty password				
		5. click on login				
5	Enter Incorrect username	1.Open Website	Login to QuillQuest	Invalid credentials message	Invalid credentials message	PASS
	and Incorrect password	2. Open admin.jsp page	http://localhost:8080/ppranjal/admin.jsp	will pop up at top of the page	will pop up at top of the page	
		3.Enter incorrect username	pranjal			
		4. Enter Incorrect password	1234567			
		5. click on login				

Test Scenario ID: 2
Tester Name: Pranjal Shinde
Test Page: User Registration Page of Online Book Store

Test ID	Scenario	Test Case steps	Test Data	Actual Output	Expected Output	Status
1	Enter the correct username	1.Open Website	Login to QuillQuest			
	email and password	2. Open registration.jsp page	http://localhost:8080/ppranjal/registration.jsp	Redirect to the login.jsp	Redirect to the login.jsp	PASS
		3.Enter valid username, email	pranjal, pran@gmail.com	page	page	
		4. Enter correct password	pranjal			
		5. click on register				
2	Enter correct username	1.Open Website	Login to QuillQuest			
	email and incorrect password	2. Open registration.jsp page	http://localhost:8080/ppranjal/registration.jsp	Invalid credentials message	Invalid credentials message	PASS
		3.Enter valid username , email	admin , pran12345@gmail.com	will pop up at top of the page	will pop up at top of the page	
		4. Enter incorrect password	12345			
		5. click on register				
3	Enter incorrect username	1.Open Website	Login to QuillQuest	Invalid credentials message	Invalid credentials message	PASS
	email and correct password	2. Open registration.jsp page	http://localhost:8080/ppranjal/registration.jsp	will pop up at top of the page	will pop up at top of the page	
		3.Enter incorrect email address	123Pranjal, jashu123@gamil.com			
		4. Enter correct password	pranjal			
		5. click on register				
4	Enter empty username	1.Open Website	Login to QuillQuest	Invalid credentials message	Invalid credentials message	PASS
	email and password both	2. Open registration.jsp page	http://localhost:8080/ppranjal/registration.jsp	will pop up at top of the page	will pop up at top of the page	
		3.Empty email address				
		4. Empty password				
		5. click on register				
5	Enter Incorrect username	1.Open Website	Login to QuillQuest	Invalid credentials message	Invalid credentials message	PASS
	email and password	2. Open registration.jsp page	http://localhost:8080/ppranjal/registration.jsp	will pop up at top of the page	will pop up at top of the page	
		3.Enter incorrect email address	puransh, prjuu12@gmail.com			
		4. Enter Incorrect password	1234567			
		5. click on register				

Test Scenario ID: 3
Tester Name: Pranjal Shinde
Test Page: User Login Page of Online Book Store Website

Test ID	Scenario	Test Case steps	Test Data	Actual Output	Expected Output	Status
1	Enter the correct user email	1.Open Website	Login to QuillQuest			
	and password	2. Open login.jsp page	http://localhost:8080/ppranjal/login.jsp	Redirect to the home.jsp	Redirect to the home.jsp	PASS
		3.Enter valid email address	pran@gmail.com	page	page	
		4. Enter correct password	pranjal			
		5. click on login				
2	Enter correct user email address	1.Open Website	Login to QuillQuest			
	and incorrect password	2. Open login.jsp page	http://localhost:8080/ppranjal/login.jsp	Invalid credentials message	Invalid credentials message	PASS
		3.Enter valid email address	pran@gmail.com	will pop up at top of the page	will pop up at top of the page	
		4. Enter incorrect password	12345			
		5. click on login				
3	Enter incorrect user email address	1.Open Website	Login to QuillQuest	Invalid credentials message	Invalid credentials message	PASS
	and correct password	2. Open login.jsp page	http://localhost:8080/ppranjal/login.jsp	will pop up at top of the page	will pop up at top of the page	
		3.Enter incorrect email address	123Pranjal			
		4. Enter correct password	pranjal			
		5. click on login				
4	Enter empty user email address	1.Open Website	Login to QuillQuest	Invalid credentials message	Invalid credentials message	PASS
	and password both	2. Open login.jsp page	http://localhost:8080/ppranjal/login.jsp	will pop up at top of the page	will pop up at top of the page	
		3.Empty email address				
		4. Empty password				
		5. click on login				
5	Enter Incorrect user email address	1.Open Website	Login to QuillQuest	Invalid credentials message	Invalid credentials message	PASS
	and Incorrect password	2. Open login.jsp page	http://localhost:8080/ppranjal/login.jsp	will pop up at top of the page	will pop up at top of the page	
		3.Enter incorrect email address	pranjal			
		4. Enter Incorrect password	1234567			
		5. click on login				

8.2 TESTING APPROACH

A testing approach refers to the systematic method or strategy used to verify and validate software or system functionality to ensure it meets the specified requirements and quality standards. There are several testing approaches, each with its own focus and objectives. Here are some common testing approaches:

Black Box Testing Approach:
Tests software functionality without knowledge of internal code structure or implementation details.
Testers focus on input-output behavior and expected functionality.
Test cases are derived from specifications and requirements.

White Box Testing Approach:
Tests software with knowledge of internal code structure and implementation details.
Tests internal logic, control flow, and data structures.
Test cases are derived from code analysis and design.

Exploratory Testing Approach:
Informal and ad-hoc testing approach where testers explore the software to uncover defects and issues.
Testers use their domain knowledge and creativity to discover bugs.
Test scenarios are not predefined, allowing for flexibility and discovery of unexpected behavior.

8.3 UNIT TESTING

Unit testing is a type of software testing that focuses on individual units or components of a software system. The purpose of unit testing is to validate that each unit of the software works as intended and meets the requirements. Unit testing is typically performed by developers, and it is performed early in the development process before the code is integrated and tested as a whole system.

Unit tests are automated and are run each time the code is changed to ensure that new code does not break existing functionality. Unit tests are designed to validate the smallest possible unit of code, such as a function or a method, and test it in isolation from the rest of the system. This allows developers to quickly identify and fix any issues early in the development process, improving the overall quality of the software and reducing the time required for later testing.

8.4 INTEGRATED TESTING

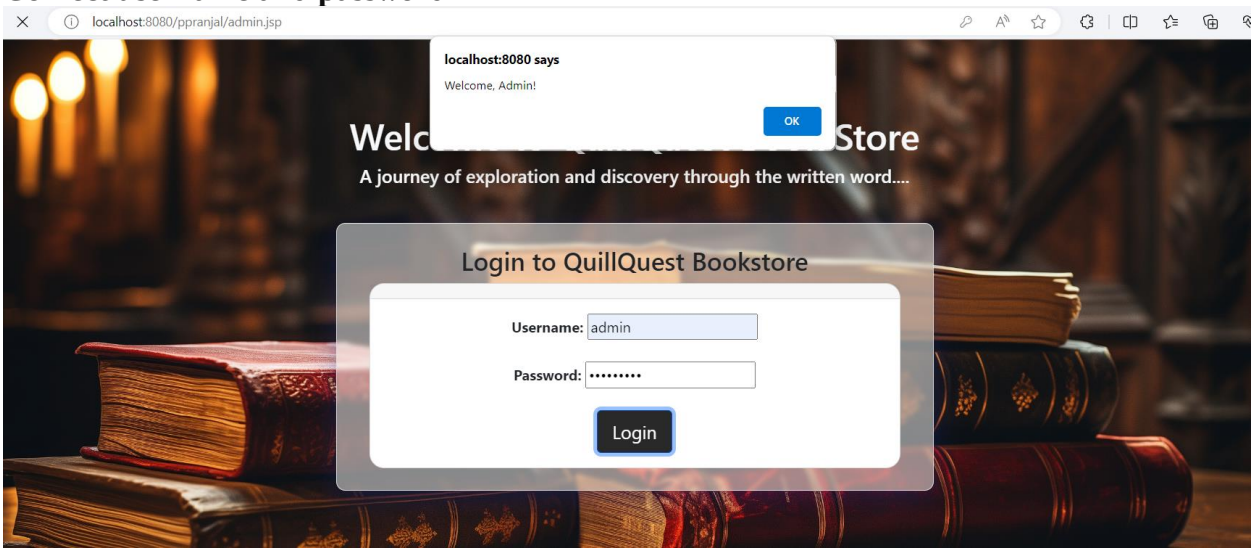
Integration testing is a software testing technique that focuses on verifying the interactions and data exchange between different components or modules of a software application. The goal of integration testing is to identify any problems or bugs that arise when different components are combined and interact with each other. Integration testing is typically performed after unit testing and before system testing. It helps to identify and resolve integration issues early in the development cycle, reducing the risk of more severe and costly problems later on.

Integration testing can be done by picking module by module. This can be done so that there should be a proper sequence to be followed. And also if you don't want to miss out on any integration scenarios then you have to follow the proper sequence. Exposing the defects is the major focus of the integration testing and the time of interaction between the integrated units.

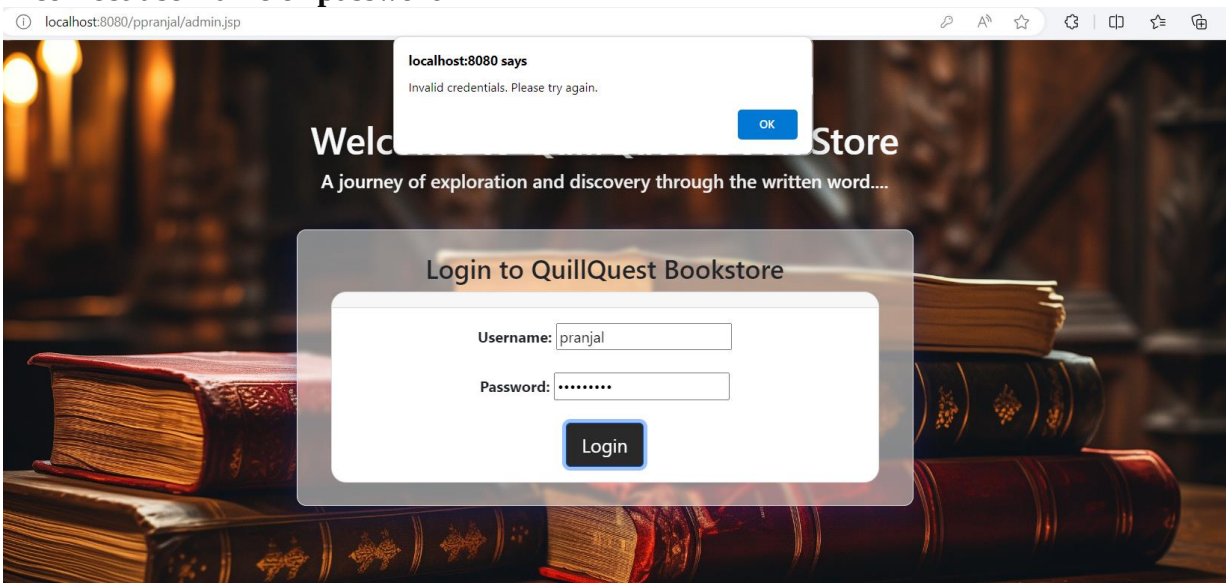
8.5TESTING OUTPUTS

Admin Login

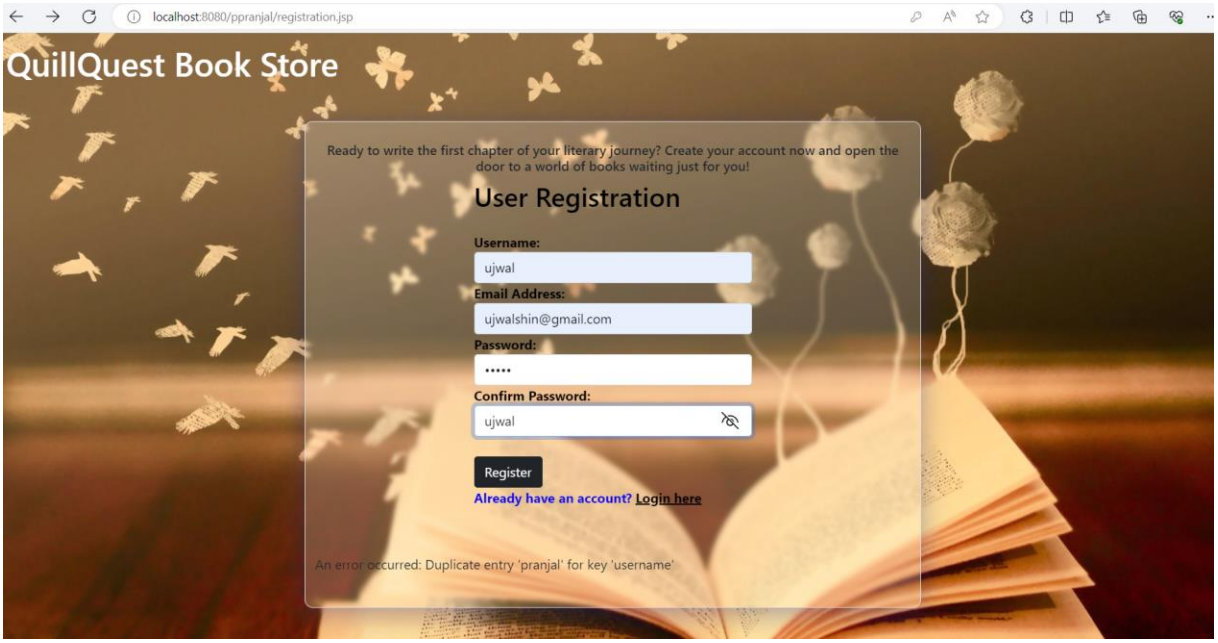
Correct username and password

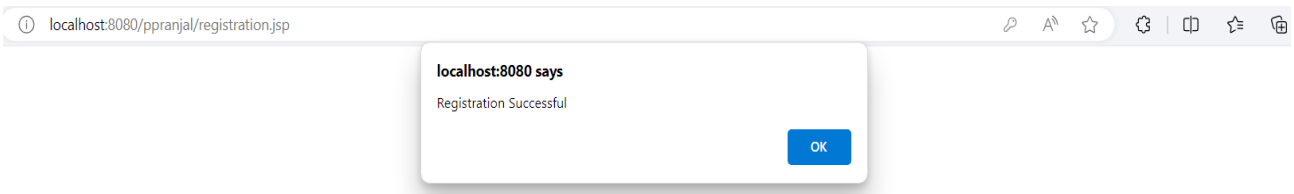


Incorrect username or password

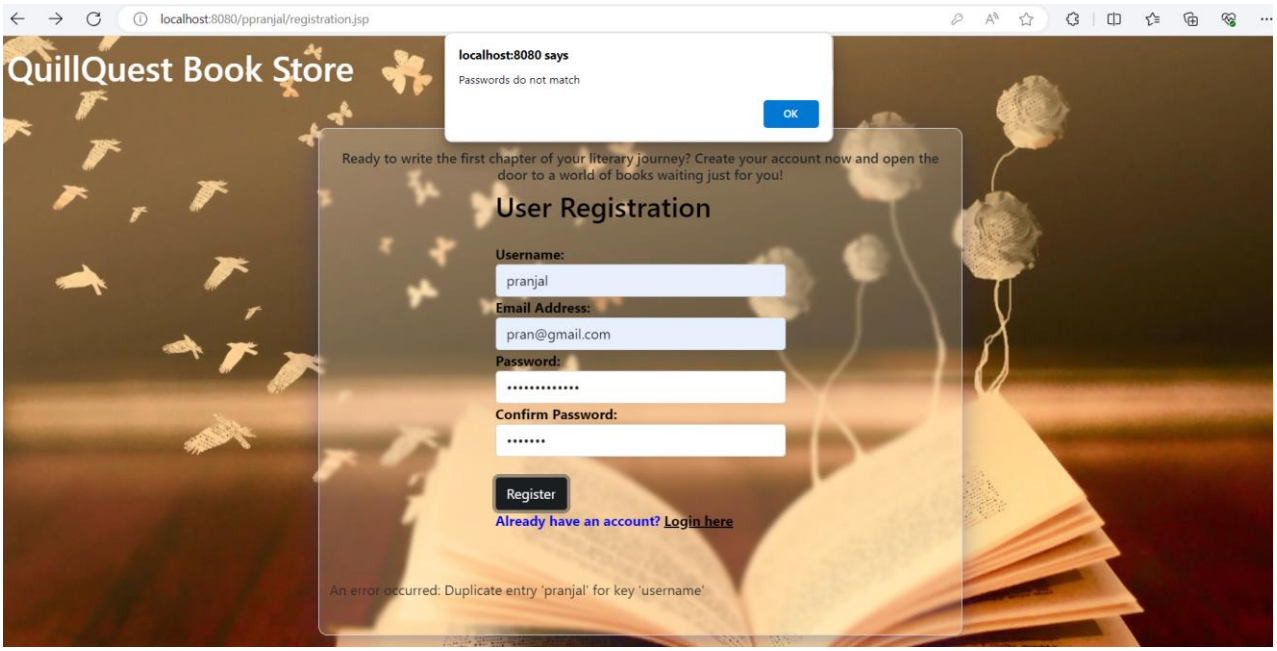


User Registration
Correct Credentials

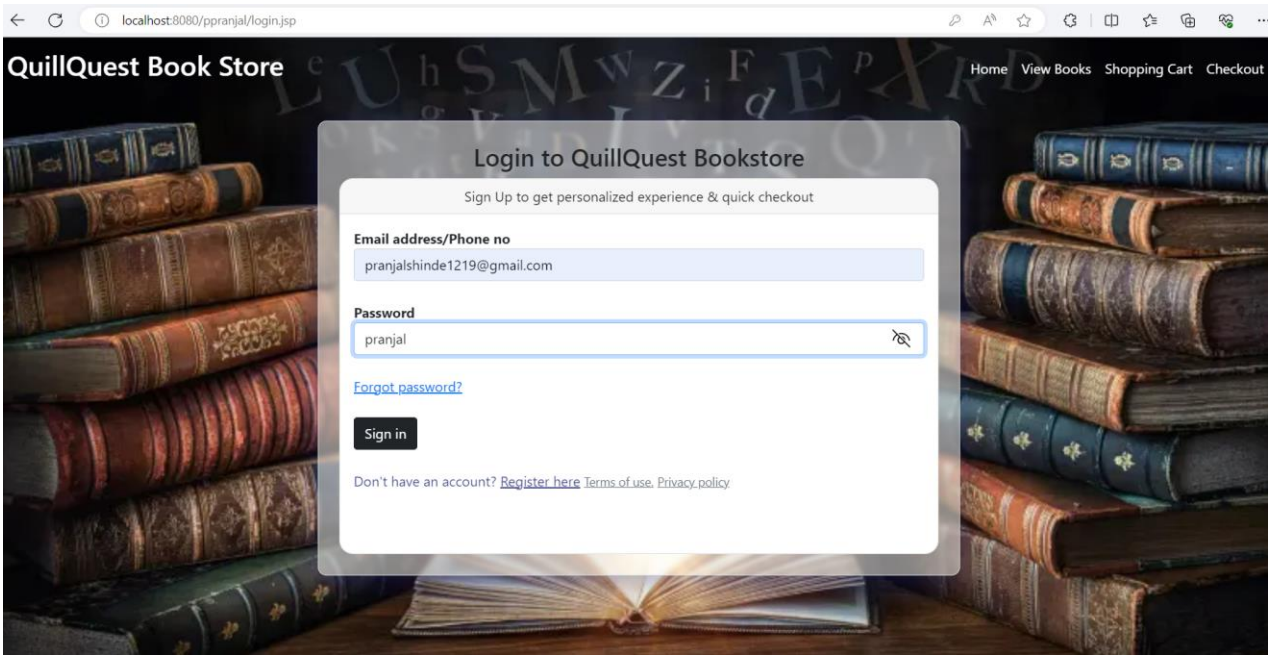




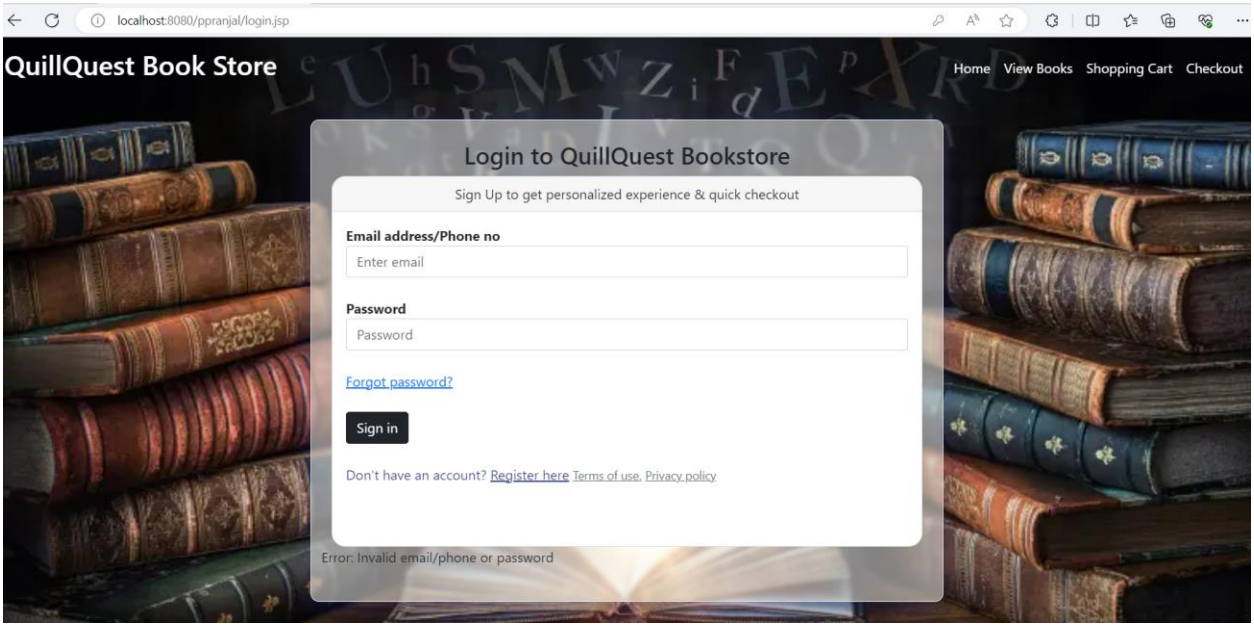
Incorrect Credentials



User Login
Registered user Correct Credentials: It will redirect to home page



Incorrect Credentials



8.6 SECURITY ISSUES

In today's world, due to huge advancements on the internet, we can find anything and everything on the internet. Need something good to eat? Order food online, and it gets delivered in a few minutes. Want to buy some clothes? Order online! Not only products, but we can also book services online and even make payments. But all this is built upon modern applications be it web or mobile. Since we are heavily dependent on these websites, we do not mind storing our personal data or even financial details like credit card numbers, etc. on the web application. But sometimes this results in a great loss in terms of data and reputation.

During the Covid-19 scenario, we have seen that the Internet is the backbone of everything, be it office meetings, online classes, virtual appointments of doctors, and a lot more. We are heavily dependent on web applications and the services and products that come with them. No physical contact has even pushed more and more sellers or service providers online. But this has also increased a huge amount of security threat that comes with it. The security of our data depends upon the website we are storing our information on. Recently, there has been a surge in security attacks, even the biggest brands couldn't escape them. A few examples of recent breaches are Microsoft Exchange (March 2021), Facebook and LinkedIn (January and March 2021), Clubhouse (April 2021), Bose (May 2021). Therefore, protecting our web applications or websites is of utmost importance, hence here I'm explaining some security risks/issues that are associated with web applications or websites so that everyone can take the necessary steps to prevent them!

Most common Website Security Issues are as follows:

1. **SQL INJECTION**

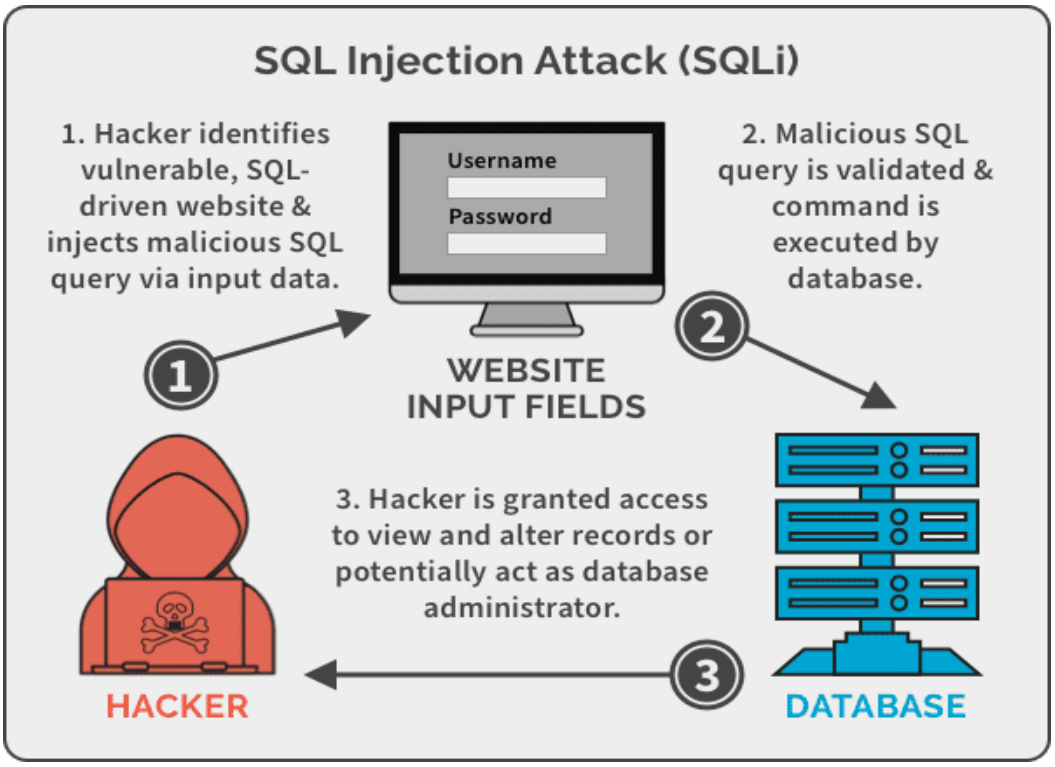
SQL injection is a technique used to extract user data by injecting web page inputs as statements through SQL commands. Basically, malicious users can use these instructions to manipulate the application's web server.

- SQL injection is a code injection technique that can compromise your database.
- SQL injection is one of the most common web hacking techniques.
- SQL injection is the injection of malicious code into SQL statements via web page input.

Web servers communicate with database servers anytime they need to retrieve or store user data. SQL statements by the attacker are designed so that they can be executed while the web server is fetching content from the application server. It compromises the security of a web application. The hacker can retrieve all the user data present in the database such as user details, credit card information, and social security numbers, and can also gain access to protected areas like the administrator portal. It is also possible to delete user data from the tables.

Nowadays, all online shopping applications and bank transactions use back-end database servers. So in case the hacker is able to exploit SQL injection, the entire server is compromised. Developers can prevent SQL Injection vulnerabilities in web applications by utilizing parameterized database queries with bound, typed parameters and careful use of parameterized stored procedures in the database.

Imperva is a comprehensive web application security platform that includes SQL injection detection capabilities. It can detect various types of SQL injection vulnerabilities, including blind SQL injection. Imperva has a user-friendly interface that allows users to manage and track vulnerabilities.



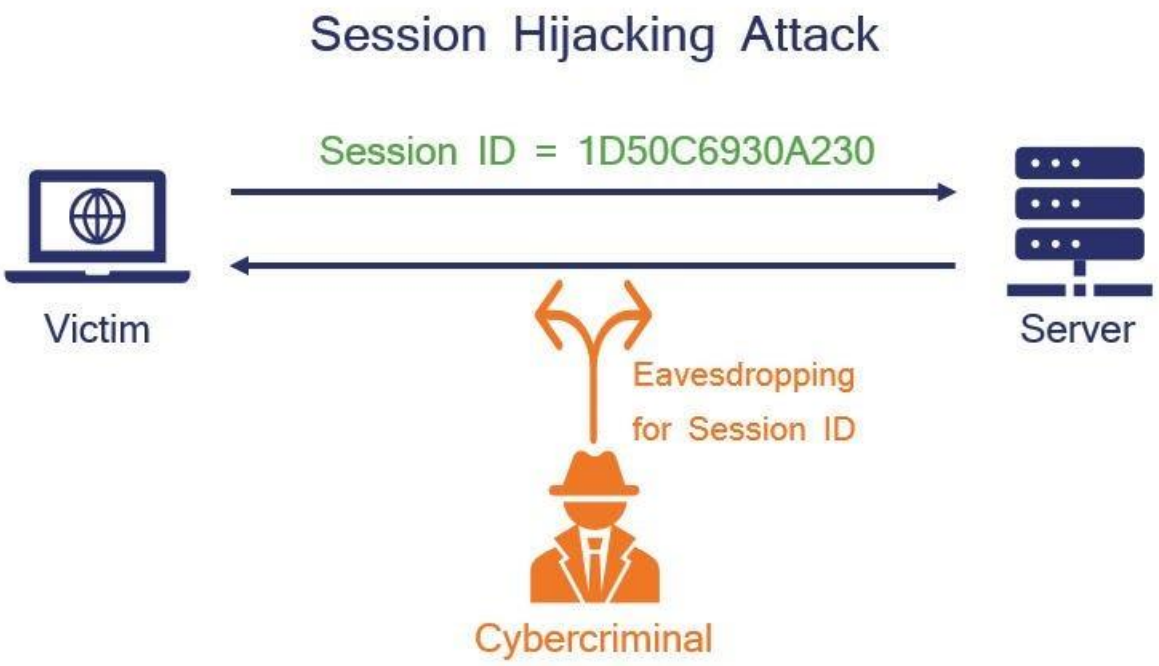
2. BROKEN AUTHENTICATION

Authentication is “broken” when attackers are able to compromise passwords, keys or session tokens, user account information, and other details to assume user identities. Due to poor design and implementation of identity and access controls, the prevalence of broken authentication is widespread.

It is a case where the authentication system of the web application is broken and can result in a series of security threats. This is possible if the adversary carries out a brute force attack to disguise itself as a user, permitting the users to use weak passwords that are either dictionary words or common passwords like “12345678”, “password” etc. This is so common because shockingly 59% of the people use the same passwords on all websites they use. Moreover, 90% of the passwords can be cracked in close to 6 hours! Therefore, it is important to permit users to use strong passwords with a combination of alphanumeric and special characters. This is also possible due to credential stuffing, URL rewriting, or not rotating session IDs.

Recommended Best Practices to Prevent Broken Authentication Attacks

Implement multifactor authentication to bypass broken session management attacks.
Enforce input validation and limited login tries.
Limit session times.
Enforce high password complexity for user accounts.
Make frequent checks for weak passwords.



3. INSECURE DIRECT OBJECT REFFERENCES

This is a classic case of trusting user input and paying the price in a resulting security vulnerability. A direct object reference means that an internal object sunch as a file or database key is exposed to the user. The problem with this is that the attacker can provide this reference and, if authorization is either not enforced (or is broken), the attacker can access or do things that they should be precluded from.

For example, the code has a download.php module that reads and lets the user download files, using a CGI parameter to specify the file name. Either by mistake or due to laziness, the developer omitted authorization from the code. The attacker can now use this to download any system files that the user running PHP has access to, like the application code itself or other data left lying around on the server, like backups.

Another common vulnerability example is a password reset function that relies on user input to determine whose password we're resetting. After clicking the valid URL, an attacker can just modify the username field in the URL to say something like "admin".
Incidentally, both of these examples are things I myself have seen appearing often "in the wild".

Prevention: Perform user authorization properly and consistently, and white-list the choices. More often than not though, the whole problem can be avoided by storing data internally and not relying on it being passed from the client via CGI parameters. Session variables is most frameworks are well suited for this purpose.

4. SENSITIVE DATA EXPOSURE

Sensitive data exposure refers to a security vulnerability where sensitive information, such as personal or financial data, is inadvertently or deliberately disclosed to unauthorized individuals or entities. This exposure can occur due to various reasons, including inadequate data protection measures, weak encryption, insecure data storage, or vulnerabilities in software or systems.

Some common examples of sensitive data exposure include:
Unencrypted Data Transmission: Sending sensitive data over unsecured channels without encryption, such as plain text email or HTTP instead of HTTPS, which can be intercepted by attackers.
Insecure Storage: Storing sensitive data in databases or files without proper access controls or encryption, making it vulnerable to unauthorized access or theft.
Poor Authentication and Authorization: Weak authentication mechanisms or improper authorization settings that allow unauthorized users to gain access to sensitive data.
Third-party Services: Entrusting sensitive data to third-party services or vendors without ensuring they have adequate security measures in place, leading to potential data breaches.
Application Vulnerabilities: Security flaws in software applications or websites that can be exploited to access sensitive data, such as SQL injection or cross-site scripting (XSS) attacks.
Insider Threats: Malicious or negligent actions by insiders, such as employees or contractors, who have legitimate access to sensitive data but misuse it or accidentally expose it.

To mitigate the risk of sensitive data exposure, organizations should implement robust security measures, such as encryption, access controls, regular security assessments, employee training on data handling best practices, and compliance with data protection regulations like GDPR (General Data Protection Regulation) or CCPA (California Consumer Privacy Act). Additionally, implementing data minimization practices and only collecting and retaining the

minimum amount of sensitive data necessary for business purposes can help reduce the potential impact of a data breach.

5. CROSS SITE REQUEST FORGERY(CSRF)

Cross-Site Request Forgery (CSRF) is a type of web security vulnerability where an attacker tricks a user into unintentionally executing actions on a web application in which the user is authenticated. This is achieved by crafting a malicious request and getting the victim to execute it while they are authenticated to the target application.

CSRF attacks can have serious consequences, such as unauthorized fund transfers, account takeover, data manipulation, or unauthorized actions performed on behalf of the victim.

To **prevent CSRF attacks**, web developers can implement several security measures:

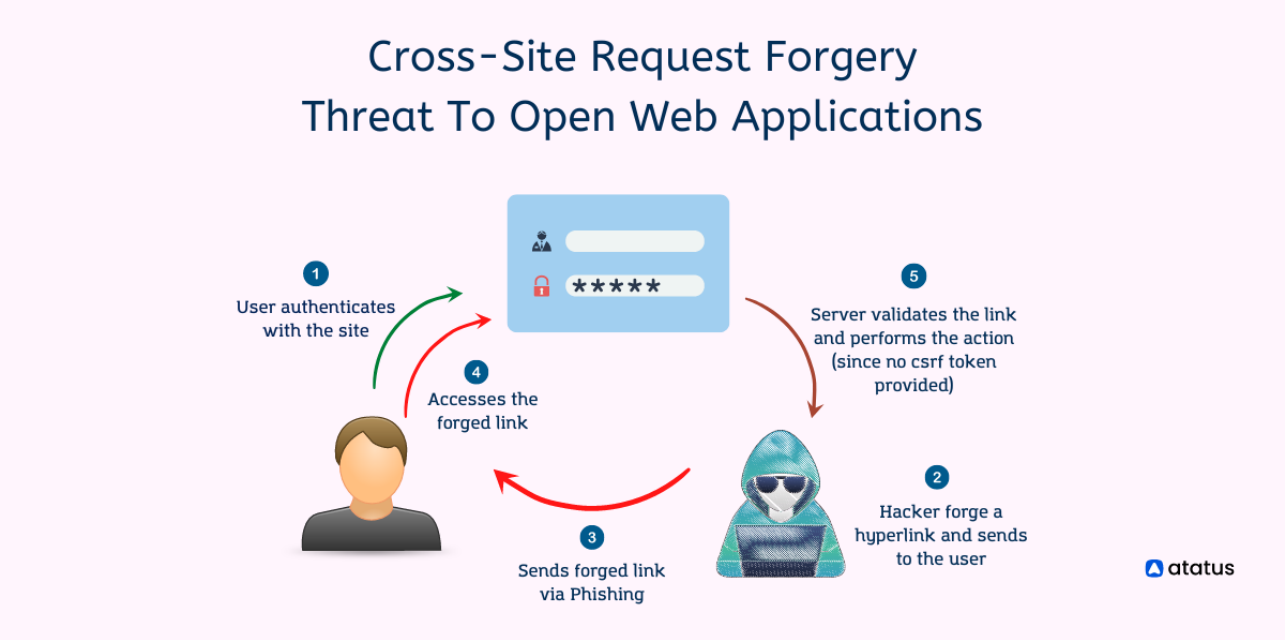
CSRF Tokens: Include unique tokens in forms or requests that are required for actions to be executed. These tokens are generated by the server and embedded in the form or request and must be submitted along with the request. Since the attacker cannot obtain this token due to the same-origin policy, they cannot craft a valid malicious request.

Same-Site Cookies: Set cookies to be same-site, meaning they are only sent in requests originating from the same site as the target application, reducing the risk of CSRF attacks.

HTTP Referer Header: Validate the HTTP Referer header to ensure that requests originate from trusted sources.

Use of POST: Design web applications to use POST requests for actions that modify state (such as changing settings or transferring funds), as POST requests are less likely to be triggered by an attacker unknowingly.

By implementing these measures, web developers can effectively mitigate the risk of CSRF attacks and enhance the security of their applications.



6. CROSS SITE SCRIPTING

Cross-Site Scripting (XSS) is a type of web security vulnerability that allows attackers to inject malicious scripts into web pages viewed by other users. These scripts can be executed in the context of the victim's browser, allowing the attacker to steal sensitive information, hijack user sessions, deface websites, or perform other malicious actions.

XSS attacks can have serious consequences, including theft of sensitive information (such as session cookies or personal data), unauthorized access to user accounts, spreading malware, defacement of websites, and more.

To **prevent XSS attacks**, web developers can implement several security measures:

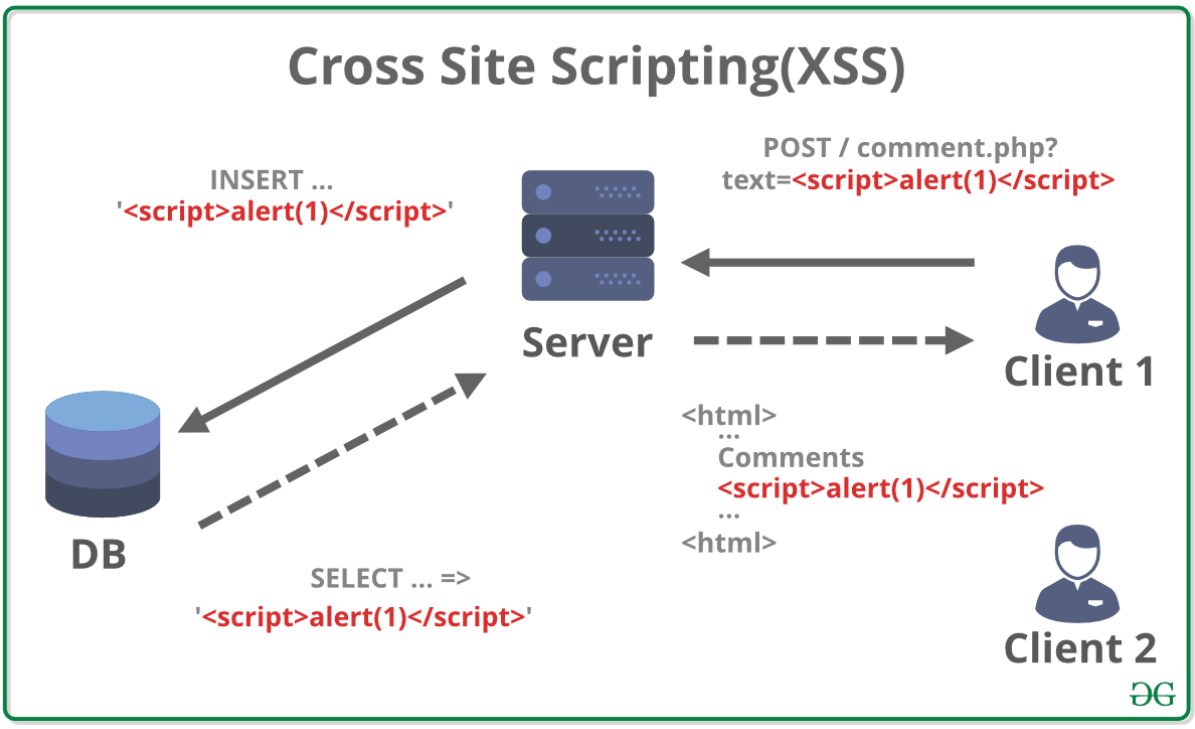
Input Validation and Sanitization: Validate and sanitize all user input to ensure that it does not contain malicious scripts or HTML tags. Input validation should be performed both on the client-side and server-side.

Content Security Policy (CSP): Implement CSP headers to restrict the sources from which certain types of content (such as scripts, stylesheets, and images) can be loaded, thereby mitigating the impact of XSS attacks.

Use of Security Libraries: Utilize security libraries and frameworks that provide built-in protections against XSS attacks, such as OWASP ESAPI or DOMPurify.

Cookie Security: Set the 'HttpOnly' flag on cookies to prevent them from being accessed by client-side scripts, reducing the risk of session hijacking via XSS attacks.

By implementing these best practices, web developers can significantly reduce the risk of XSS vulnerabilities and enhance the security of their web applications. Regular security testing and code reviews are also crucial to identifying and mitigating XSS vulnerabilities effectively.



THE IMPORTANCE OF ENCRYPTION

Encryption plays a crucial role in ensuring the confidentiality, integrity, and authenticity of sensitive data in various contexts. With using more and more technologies in our lives, we are generating large amounts of data, a great share of which is sensitive data. If someone else gets hold of that data you may be at risk of financial fraud or identity theft. Encryption helps protect sensitive information from unauthorized access by converting it into an unreadable format (ciphertext) that can only be decrypted by authorized parties with the appropriate decryption key. This ensures that even if data is intercepted or compromised, it remains unintelligible to anyone without the decryption key.

Encryption safeguards data at rest (stored data) and in transit (data being transmitted over networks) from unauthorized viewing or tampering. This is especially important for sensitive data such as personal information, financial records, and intellectual property, helping organizations comply with data protection regulations and maintain customer trust. It enhances the security and privacy of communication channels, preventing eavesdropping, interception, and man-in-the-middle attacks. It secures communications over the internet, email, messaging platforms, and other communication channels, ensuring that sensitive information remains private and secure.

Many regulatory frameworks and industry standards, such as GDPR, HIPAA, PCI DSS, and FERPA, mandate the use of encryption to protect sensitive data and ensure compliance with data protection laws. Encryption helps organizations meet these legal and regulatory requirements by safeguarding sensitive data from unauthorized access and data breaches. Encryption mitigates the risks associated with data breaches, cyberattacks, and unauthorized access by making stolen or compromised data unusable to attackers without the decryption key. It serves as an essential layer of defense in multi-layered security strategies, reducing the potential impact of security incidents and data breaches. Implementing encryption demonstrates a commitment to data security and customer privacy, enhancing trust and reputation among customers, partners, and stakeholders. It reassures users that their sensitive information is being handled securely and responsibly, strengthening relationships and fostering confidence in products and services. Encryption helps protect intellectual property, trade secrets, and proprietary information from theft, espionage, and unauthorized disclosure. By encrypting valuable assets, organizations can safeguard their competitive advantage and preserve the confidentiality of sensitive business data.

In summary, encryption is essential for safeguarding sensitive data, maintaining compliance with regulations, mitigating security risks, and fostering trust and confidence in digital communications and transactions. It is a fundamental tool in cybersecurity strategies to protect confidentiality, integrity, and authenticity, ensuring the secure transmission and storage of information in an increasingly interconnected and data-driven world.

1. (TRIPLE) DES

Triple DES (Data Encryption Standard) is a symmetric-key block cipher algorithm used to encrypt data. It is an enhancement of the original DES encryption algorithm, which uses a single key. Triple DES applies the DES algorithm three times, using three different keys, hence the name "triple." This process increases the key length and the overall security of the encryption. Here's how Triple DES works:

Key Expansion: In Triple DES, three keys are used: Key1, Key2, and Key3. Key1 is used for the first DES encryption, Key2 for decryption, and Key3 for encryption again.

Encryption: The plaintext is first encrypted using Key1 with the DES algorithm, producing ciphertext1. Then, the ciphertext1 is decrypted using Key2 with the DES algorithm, resulting in an intermediate plaintext. Finally, this intermediate plaintext is encrypted again using Key3 with the DES algorithm, producing the final ciphertext.
Decryption: To decrypt the ciphertext, the process is reversed. The ciphertext is first decrypted using Key3, then encrypted using Key2, and finally decrypted using Key1.

Triple DES provides a significantly stronger level of security compared to DES due to its longer key length. While DES uses a 56-bit key, Triple DES effectively uses either a 112-bit key or a 168-bit key, depending on whether two or three keys are used. However, Triple DES is slower and less efficient compared to modern encryption algorithms such as AES (Advanced Encryption Standard).

Despite its enhanced security over DES, Triple DES is considered obsolete for new applications due to its slower speed and susceptibility to certain cryptographic attacks. AES is now the preferred choice for symmetric-key encryption in most situations due to its higher security, faster speed, and support for a wider range of key lengths.

2. RSA

RSA (Rivest-Shamir-Adleman) is a widely used asymmetric cryptographic algorithm for secure data transmission, digital signatures, and key exchange. It is named after its inventors: Ron Rivest, Adi Shamir, and Leonard Adleman, who introduced it in 1977.

Asymmetric cryptography involves the use of a pair of keys: a public key and a private key. These keys are mathematically related such that data encrypted with one key can only be decrypted with the other key in the pair. In RSA, the public key is used for encryption and verifying digital signatures, while the private key is used for decryption and creating digital signatures.
Here's how RSA works:
Key Generation: To generate an RSA key pair, two large prime numbers (p and q) are selected. Their product (n = p * q) forms the modulus for both the public and private keys. Additionally, the public exponent (e) is chosen, usually a small prime number like 65537, which is relatively prime to (p-1)(q-1). The private exponent (d) is calculated such that (d * e) mod ((p-1)(q-1)) = 1.
Encryption: To encrypt a message (plaintext), the sender uses the recipient's public key (n, e). The plaintext is raised to the power of the public exponent (e) modulo the modulus (n), resulting in the ciphertext.
Decryption: To decrypt the ciphertext, the recipient uses their private key (d). The ciphertext is raised to the power of the private exponent (d) modulo the modulus (n), recovering the original plaintext.
Digital Signatures: To create a digital signature, the sender hashes the message, then encrypts the hash with their private key. The recipient can verify the signature by decrypting it with the sender's public key and comparing the decrypted hash with the hash of the received message.

RSA is widely used in various applications, including secure communication protocols like SSL/TLS, SSH, and PGP, as well as digital certificates for authentication and data integrity. However, RSA's security relies on the difficulty of factoring large numbers, and with advances in computing power and cryptographic attacks, longer key lengths are required to maintain security. As a result, RSA is gradually being replaced by algorithms like Elliptic Curve Cryptography (ECC) for many applications, due to its smaller key sizes and equivalent security.

CHAPTER – 9 (MAINTENANCE)

9.1 MODIFICATIONS AND IMPROVEMENTS

Modifications and improvements for an online bookstore website project can be approached from various angles to enhance its functionality, usability, security, and overall user experience. Here are some key points to consider for modifications and improvements:

User Interface Enhancements:
Implement a modern and responsive design to ensure compatibility with different devices and screen sizes.
Improve navigation by organizing books into categories, providing search and filter options, and enhancing the browsing experience.
Add features such as book recommendations, user reviews, and personalized recommendations based on user preferences and past purchases.

Performance Optimization:
Optimize page loading speed by minimizing HTTP requests, leveraging browser caching, and optimizing images and other assets.
Implement lazy loading techniques to load content dynamically as users scroll, reducing initial load times.
Use content delivery networks (CDNs) to serve static assets and improve global performance.

Security Enhancements:
Implement secure authentication and authorization mechanisms to protect user accounts and sensitive information.
Use HTTPS protocol to encrypt data transmitted between the server and the client, ensuring data confidentiality and integrity.
Implement measures to prevent common security threats such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).

Payment Gateway Integration:
Integrate popular payment gateways to provide users with secure and convenient payment options.
Ensure compliance with PCI-DSS (Payment Card Industry Data Security Standard) requirements to safeguard payment card data.

Inventory Management:
Implement an inventory management system to track book availability, stock levels, and sales.
Provide real-time updates on product availability and notify users when items are back in stock.

Order Management and Tracking:
Enhance the order management system to track order statuses, manage shipping and delivery, and provide order tracking functionality.
Send order confirmation emails and notifications to keep users informed about their order status.

Customer Feedback and Support:
Implement a feedback system to collect user reviews, ratings, and suggestions for improving the website.
Provide multiple channels for customer support, such as live chat, email, and phone support, to address user inquiries and concerns promptly.

Analytics and Reporting:
Integrate web analytics tools to track user behavior, monitor website performance, and gain insights into customer preferences and trends.
Generate reports on sales performance, customer demographics, and popular book categories to inform business decisions and marketing strategies.

Continuous Improvement and Updates:
Regularly update the website with new features, content, and improvements based on user feedback and market trends.
Conduct usability testing and gather user feedback to identify areas for further optimization and refinement.

By addressing these modifications and improvements, you can create a more robust, user-friendly, and competitive online bookstore website that meets the evolving needs and expectations of customers.

CHAPTER – 10 (CONCLUSION)

9.1 CONCLUSION

In conclusion, the development of the QuillQuest Online Bookstore website has been a significant endeavor aimed at providing book enthusiasts with a seamless and enriching shopping experience in the digital realm. Through meticulous planning, thoughtful design, and diligent implementation, the project has successfully created a platform where users can explore, discover, and purchase their favorite books with ease. The integration of intuitive user interfaces, secure payment gateways, and comprehensive inventory management systems ensures that users can navigate the website confidently while enjoying a wide selection of literary offerings.

Looking ahead, continuous improvement and adaptation will be key to maintaining the website's relevance and competitiveness in the dynamic landscape of online retail. By embracing user feedback, staying abreast of emerging technologies, and refining existing features, the QuillQuest Online Bookstore aims to evolve into a trusted destination for book lovers worldwide. With a commitment to excellence and innovation, this project serves as a testament to the potential of digital platforms to revolutionize the way we engage with literature, fostering a community of avid readers and lifelong learners in the digital age.

9.2 FUTURE ENHANCEMENT

Future enhancements for this project can be pursued to further improve user experience, expand functionality, and stay competitive in the evolving online retail landscape. Here are some potential enhancements:

Personalized Recommendations: Implement machine learning algorithms to analyze user browsing and purchase history, allowing the website to offer personalized book recommendations tailored to each user's interests and preferences.
Social Integration: Integrate social media features to enable users to share their favorite books, reviews, and reading lists with their friends and followers. This can enhance user engagement and promote word-of-mouth marketing.
Virtual Book Clubs: Create a platform for users to join virtual book clubs, participate in discussions, and connect with other readers who share similar literary interests. This can foster a sense of community among users and encourage ongoing engagement with the website.
Expanded Content Formats: Diversify the content offerings by including audiobooks, e-books, and other digital formats in addition to traditional print books. This caters to the preferences of users who prefer alternative reading formats.
Subscription Services: Introduce subscription-based services such as book rental, lending libraries, or monthly book boxes, providing users with access to a curated selection of books at a fixed fee. This can offer added value and convenience to customers while generating recurring revenue for the website.

By implementing these future enhancements, the QuillQuest Online Bookstore can continue to innovate and evolve, providing an engaging and fulfilling experience for book enthusiasts while staying ahead of the curve in the competitive online book retail market.

CHAPTER – 11 (REFERENCES)

The decision to develop an online bookstore website using JSP stemmed from a personal interest in the subject, supported by prior knowledge gained through academic studies in Java, Advanced Java, Java Frameworks, and web development. Initial inspiration was drawn from personal reading habits, prompting further research on the topic via Google to gather primary information. Leveraging expertise in Java and related technologies acquired through the curriculum, the choice to utilize JSP for website development was a natural fit. This comprehensive approach, combining personal interest, academic background, and technical skills, guided the development process towards creating a meaningful and engaging online platform for book enthusiasts.

Following are some websites that helped me to develop Online Bookstore Project:

- https://nevonprojects.com/online-book-store-project/
- https://youtu.be/3cDcUtY0uLQ?si=HQmJ4SZ6Pt4F4upN
- https://youtu.be/3cDcUtY0uLQ?si=DpP0pTIBQMKe7sfo
- https://youtu.be/oiRq01AMeSA?si=l2jjgyY3Q9rYVd0E
- https://chat.openai.com/c/19309176-3aa3-4162-8419-62f512760837

Thank You!!!