A black background with blue text

Description automatically generated

Enterprise Applications Development:

Online Bookstore

Talgeri Rhea Ramchandra G2203675J

# Introduction

This report is to outline the development of an ecommerce website, more specifically, an online bookstore. Although an ecommerce website is a common project that has been done many times before, this project will be focused on the learning and knowledge gained from the development of the project.

The objectives of the online bookstore, is to provide endusers a high-quality, convenient, and personalized shopping experience for purchasing books and related products.

The importance and relevance of ecommerce in today's market are undeniable, driven by several key factors:

Convenience: Ecommerce offers unparalleled convenience to consumers, allowing them to shop from the comfort of their homes or on the go using mobile devices. This convenience factor is particularly appealing in today's fast-paced society where time is a precious commodity.

Global Reach: Ecommerce transcends geographical boundaries, enabling businesses to reach customers beyond their local markets. This global reach opens up immense growth opportunities for businesses of all sizes, from small startups to multinational corporations.

24/7 Availability: Unlike traditional brick-and-mortar stores with fixed operating hours, ecommerce platforms are accessible 24/7, providing customers with the flexibility to shop whenever it's convenient for them. This round-the-clock availability caters to the diverse needs and schedules of modern consumers.

# Project Planning and Scope

During the initial project planning, the key objectives were to materialise a clear project idea, which would include a lot of the technolgies that were learnt during the semester.

A detailed project timeline was created, outlining key milestones and deliverables for each phase of the development process. This timeline served as a roadmap for tracking progress, managing dependencies, and ensuring timely completion of the project.

## Key features and functionalities

1. User Registration and Authentication:

Allow users to create accounts and log in securely.

Enable social media login options for convenience.

2. Browsing and Search:

User-friendly interface for browsing book names

3. Product Pages:

Detailed product descriptions

4. Shopping Cart and Checkout:

Add books to the shopping cart for easy purchase.

Ability to edit quantities and remove items from the cart.

5. Account Management:

View order history

6. Inventory Management:

Real-time inventory updates

# Technology Stack

1. Backend Framework:

Spring MVC: Used for building the backend infrastructure, handling HTTP requests, and implementing the Model-View-Controller (MVC) architecture in Java.

1. Programming Language:

Java: Utilized for server-side development, business logic implementation, and integration with external services and databases.

1. Frontend Framework:

JSP (JavaServer Pages): Employed for generating dynamic web pages on the client side, rendering data from the backend, and facilitating interaction with users.

1. Database Management System:

MySQL: Chosen as the relational database management system (RDBMS) for storing and managing structured data related to products, users, orders, and other aspects of the ecommerce platform.

This technology stack leverages the strengths of each component to deliver a robust, scalable, and efficient ecommerce solution. Spring MVC provides a flexible and modular framework for backend development, while Java ensures cross-platform compatibility and robustness. JSP facilitates the creation of dynamic web interfaces, enabling seamless interaction with users. MySQL serves as a reliable and performant database backend, handling data storage and retrieval with efficiency. Overall, this technology stack forms a cohesive ecosystem for developing and deploying the ecommerce project effectively.

# Development Process

Utilizing iterative development for the online bookstore project presents numerous advantages within software development:

Incremental Progress: Employing iterative development facilitates the breakdown of the project into smaller, more manageable portions or iterations. Each iteration concentrates on delivering specific features or functionalities, allowing for gradual advancement towards overarching project objectives.

Flexibility and Adaptability: Iterative development fosters flexibility and adaptability in response to evolving requirements, priorities, and market dynamics. As the project progresses through iterations, adjustments to the project plan can be made to accommodate new insights and challenges.

Risk Mitigation: Fragmenting the project into iterations aids in mitigating risks associated with extensive development efforts. Addressing critical features early in the process enables the identification and mitigation of potential risks and issues, contributing to smoother project execution.

Continuous Improvement: Iterative development cultivates a culture of ongoing improvement by leveraging insights gained from each iteration. This iterative feedback loop fosters innovation, efficiency, and overall project success over time.

In the context of the online bookstore project, iterative development facilitates the delivery of a comprehensive and feature-rich ecommerce platform while remaining adaptable to user needs and market trends. By iteratively refining and enhancing the platform based on feedback and insights gathered throughout the development process, the final product can surpass customer expectations and deliver substantial value to the business.

# Backend Development

Server-side programming languages and frameworks

# Frontend Development

The frontend of the application is primarily developed using JSP (JavaServer Pages), which seamlessly integrates Java code with HTML to create dynamic web pages. JSP allows for the incorporation of server-side logic directly into web pages, enabling the generation of content based on data retrieved from the backend.

Additionally, JavaScript has been employed to enhance the functionality of the frontend. JavaScript, a client-side scripting language executed by web browsers, enables manipulation of the webpage's structure (DOM) and response to user interactions in real-time. JavaScript functionalities have been integrated to perform tasks such as updating quantities within the shopping cart, managing item removal, and providing interactive elements like confirmation dialogs.

Through the integration of both JSP and JavaScript, the frontend delivers dynamic content and interactive features, contributing to an improved user experience and enhanced usability of the application.

# Database Structure and Entity Relationship Diagram

Entity-Relationship Diagram, is a graphical representation of the entities and their relationships to each other within a database. It's a visual tool used during the database design phase to illustrate the logical structure of the database. In an ERD, entities are represented as rectangles, relationships as lines connecting these rectangles, and attributes as ovals or ellipses connected to their respective entities. It helps to visualize how data is organized and how different entities are connected to each other within a database schema. The image below shows an ERD of the database doe the Online Bookshop.

A screenshot of a computer screen

Description automatically generated

The following is the SQL script that was used to create the databased. Due to the iterative development methodology, the tables were not all created at the same time.

SQL Script

|  |
| --- |
| CREATE TABLE users (  userId VARCHAR(255) NOT NULL PRIMARY KEY,  password VARCHAR(255) NOT NULL,  contactNo VARCHAR(255),  email VARCHAR(255) NOT NULL UNIQUE,  name VARCHAR(255) NOT NULL  );  CREATE TABLE book (  bookId INT NOT NULL PRIMARY KEY AUTO\_INCREMENT,  name VARCHAR(255),  description TEXT,  price DECIMAL(10,2)  );  CREATE TABLE orders (  orderId BIGINT NOT NULL PRIMARY KEY,  userId VARCHAR(255) NOT NULL,  amount DECIMAL(10,2) NOT NULL,  createDateTime TIMESTAMP,  FOREIGN KEY (userId) REFERENCES users(userId)  );  CREATE TABLE orderbook (  orderBookId INT NOT NULL AUTO\_INCREMENT,  orderId BIGINT NOT NULL,  bookId INT NOT NULL,  quantity INT NOT NULL,  PRIMARY KEY (orderBookId),  FOREIGN KEY (orderId) REFERENCES orders(orderId),  FOREIGN KEY (bookId) REFERENCES book(bookId)  );  CREATE TABLE cartitem (  userId VARCHAR(255) NOT NULL,  bookId INT NOT NULL,  quantity INT NOT NULL,  PRIMARY KEY (userId, bookId),  FOREIGN KEY (userId) REFERENCES users(userId),  FOREIGN KEY (bookId) REFERENCES book(bookId)  ); |

Each table and attribute in this SQL script for an online bookstore serves a specific purpose in capturing and managing the data related to the business operations. Here's why each of them is important:

Users Table (users):

userId: A unique identifier for each user, crucial for identifying and distinguishing users in the system.

password: Stores the password for user authentication, ensuring secure access to user accounts.

contactNo: Stores the contact number of users, which can be important for communication purposes.

email: Unique email addresses for user identification and communication.

name: Stores the name of the users, essential for personalization and identification purposes.

Book Table (book):

bookId: A unique identifier for each book, necessary for referencing and identifying books uniquely.

name: Stores the name/title of the book, allowing users to search and identify books easily.

description: Provides a description of the book, helping users to understand the content.

price: Stores the price of the book, essential for pricing and financial transactions.

Orders Table (orders):

orderId: Unique identifier for each order, crucial for tracking and managing orders.

userId: References the user who placed the order, establishing a relationship between orders and users.

amount: Stores the total amount of the order, essential for financial transactions and order processing.

createDateTime: Records the timestamp when the order was created, helpful for order tracking and analysis.

OrderBook Table (orderbook):

orderBookId: Unique identifier for each entry in the order book, facilitating the management of individual order items.

orderId: References the order to which the book belongs, establishing a relationship between orders and books.

bookId: References the book in the order, establishing a relationship between orders and books.

quantity: Stores the quantity of each book ordered, crucial for inventory management and order fulfillment.

CartItem Table (cartitem):

userId: References the user who added the item to the cart, establishing a relationship between users and cart items.

bookId: References the book added to the cart, establishing a relationship between books and cart items.

quantity: Stores the quantity of each book in the cart, crucial for managing the user's shopping cart and order preparation.

Overall, these tables and attributes collectively form the database schema for an online bookstore, enabling efficient management of users, books, orders, and shopping carts, which are fundamental components of the business operations.

# Results and Challenges

There were a few challenges faced during the developement of this projects.

Initially, I faced difficulties with the navigation of the application due to issues with href request mapping. Mapping URLs to controller methods using annotations like @RequestMapping or @GetMapping was confusing at first, leading to inconsistent URL routing. This caused some links to not direct users to the correct pages, which disrupted the overall user experience.

Additionally, understanding when and where to use annotations within the Spring MVC framework was challenging. With a wide range of annotations available, such as @Controller, @RequestMapping, @GetMapping, and @PostMapping, it was tricky to choose the right one for each situation. Figuring out where to apply annotations to define controllers, manage request mappings, and handle RESTful endpoints required experimentation and a thorough understanding of the framework.

One significant challenge I encountered during the development process was the lack of prior experience with Spring MVC. As I delved into Spring MVC development, I faced difficulties troubleshooting certain issues that arose along the way. The learning curve associated with understanding the intricacies of Spring MVC architecture and its various components presented a hurdle in efficiently identifying and resolving issues.

One persistent issue I faced was related to the stability of the Tomcat server used for hosting the application. Despite my best efforts to configure and optimize the server environment, I experienced intermittent halts and runtime errors. These server interruptions occurred randomly, disrupting my development workflow and impacting productivity.

The root cause of these server issues proved elusive, requiring extensive investigation and debugging efforts on my part. I meticulously reviewed server logs, examined configuration settings, and conducted stress tests to pinpoint the underlying issues. However, the sporadic nature of the problems made it challenging to replicate and diagnose them definitively.

# Future Enhancements

One of the areas for improvement is increasing the security of the applications. One future enhancement I would implement is encrypting the user data table as it contains sensitive information such as password and Name.

I would also implement guest orders, where end users would not need to login to the website to order books. This could potentially improveme sales for the websites are users would feel that they dont need to go through the hassle of creating an account to buy books.

The final future enhancement that could be implemented is a AI based recommendations for end users, that uses the user's search and order history.

12. Conclusion

Summary of key findings and achievements

Overall impact and significance of the ecommerce project

Final thoughts and recommendations

13. References

Any external resources or references cited in the report