BINUS UNIVERSITY

BINUS INTERNATIONAL

Final Project Cover Letter

(Group Work)

Student Information:	Surname:	Given Name:	Student ID Number:
1.	Kusnadi	Clarissa Audrey Fabiola	2602118490
2.	-	Jeffrey	2602118484
3.	Munthe	Priscilla Abigail	2602109883

Course Code : COMP6799001 **Course Name :** Database Technology

Class : L3AC **Lecturer** : Dr. Raymond Bahana, ST., M. Sc

Type of Assignment : Final Project Report

Submission Pattern

Due Date : 14 January 2024 **Submission Date :** 10 January 2024

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer's instructions.
2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

Plagiarism/Cheating

BiNus International seriously regards all forms of plagiarism, cheating, and collusion as academic offences which may result in severe penalties, including loss/drop of marks, course/class discontinuity, and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

Declaration of Originality

By signing this assignment, I understand, accept, and consent to BiNus International terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student:

Clarissa Audrey Fabiola Kusnadi

Jeffrey

Priscilla Abigail Munthe

Table of Contents

Project Details

Project Name: Literarium. Derived from the words “literary” and “-arium”. The term “literary” pertains to the realm of literature or written compositions, whereas the suffix “-arium” is commonly used to denote a place.

Project Topic: Book Management System

Group Name: AJA

Team Members:

- 2602118490 - Clarissa Audrey Fabiola Kusnadi - L3AC
- 2602118484 - Jeffrey - L3AC
- 2602109883 - Priscilla Abigail Munthe - L3AC

This project serves as a fundamental requirement for successfully completing the Database Technology course. Its core objective is to equip students with the skills necessary for designing and implementing a comprehensive database system that addresses a practical real-world challenge. Within the scope of this project, students are granted the freedom to choose their preferred topic. In our case, we have opted for the Book Management System as our project focus.

Problem Descriptions and Work Division

Summary

Our proposed solution "Literarium" Book Management System aims to change the way traditional bookstores manage their manual processes and keep their customer records. This manual approach is time-consuming for both customers and staff, leading to inconvenience and potential sales losses due to lengthy wait times. The reliance on traditional file systems for inventory management further escalates the problem, resulting in errors and additional costs.

To address these challenges effectively, our proposed database system offers a comprehensive solution. The automation provided by the database system will significantly reduce the operational duration, allowing customers to find and purchase books more quickly. The structured organization of data ensures accurate record-keeping, minimizing errors, elimitaing the need for excessive personnel, and ultimately reducing the costs. The transition to a database system is essential for streamlining processes, improving efficiency, and enhancing the communication between both customers and administrators.

Our "Literarium" Book Management System stands as a transformative solution that provides comprehensive features and user-friendly UI. It leverages the power of a well-designed database to revolutionize the bookstore's operations. Customers can effortlessly search for books, read reviews, and make purchases, all while receiving real-time updates on availability and Purchase status. Meanwhile, administrators can manage book listings, customer accounts, and Purchases with ease, leveraging the system's reporting and analytics capabilities for data-driven decision-making.

Task Division

	Database Design	Database Query	Programming	UI Design	Report
Abigail	✅	✅	✅		✅

Audrey	✅		✅	✅	✅
Jeffrey	✅	✅	✅		✅

Database Design

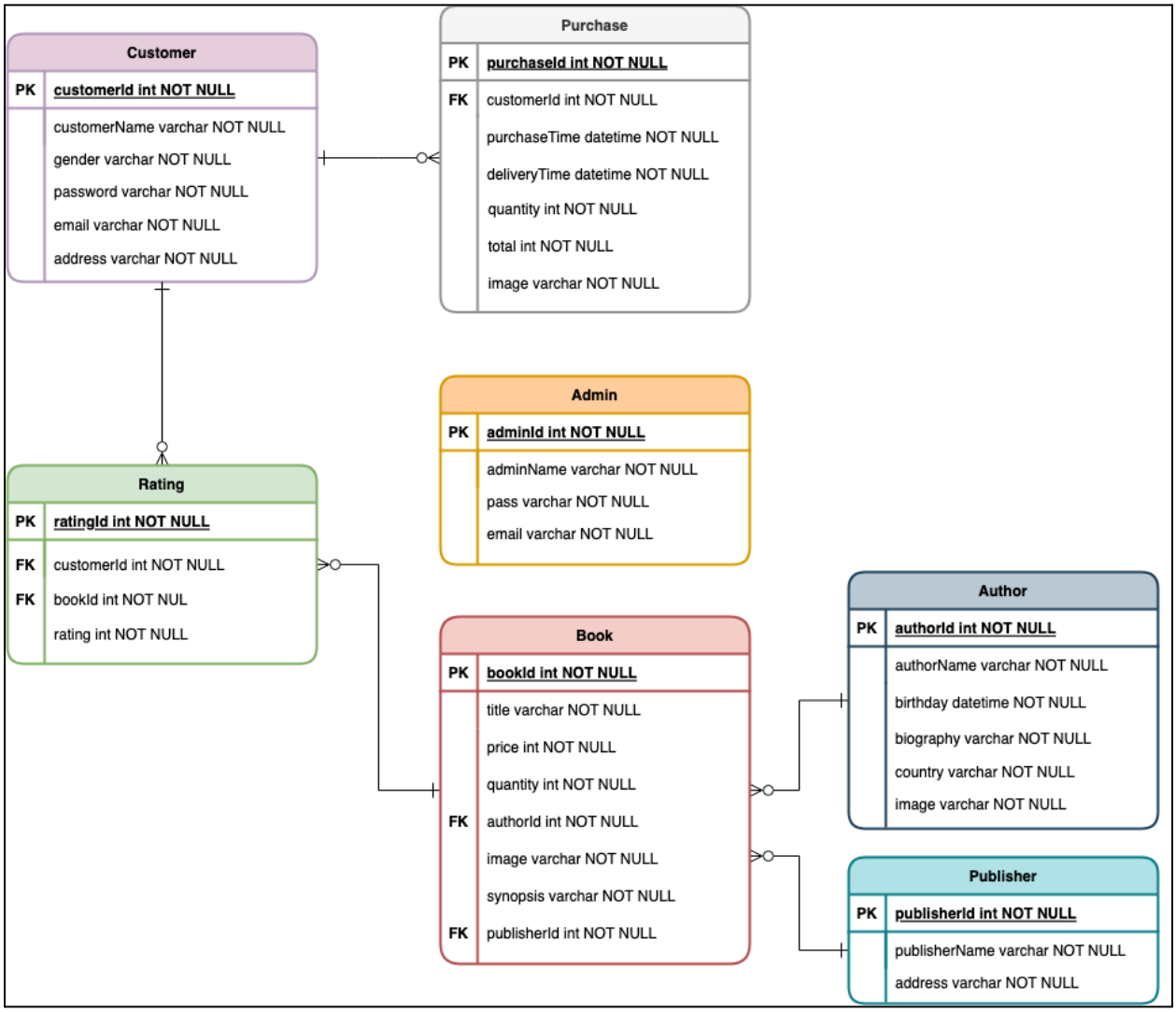
Table Description

- Customer Table:
    - Attributes:
        - customerId: Unique identifier for each customer. (Primary Key)
        - customerName: Name of the customer - NOT NULL
        - gender: Gender of the customer.
        - pass: Customer's password (Note: For security reasons, it's advisable to store password hashes instead of plain text).
        - email: Email address of the customer. - unique constraints
        - address: Physical location or address of the customer.
    - Purpose: Stores information about customers who use the system, including their identification, personal details, and login credentials. Necessary for managing user accounts and authentication.
- Book Table:
    - Attributes:
        - bookId: Unique identifier for each book. (Primary Key)
        - title: Title of the book.
        - price: Price of the book.
        - quantity: Number of copies available.
        - authorId: References the id in the Author table (Foreign Key).
        - image: Image reference or path for the book cover.
        - synopsis: Brief overview or synopsis of the book.
        - publisherId: References the id in the Publisher table (Foreign Key).
    - Purpose: Stores details about available books in the system, including their pricing, availability, author information, and other descriptive data for display or purchase.
- Author Table:
    - Attributes:
        - authorId: Unique identifier for each author. (Primary Key)

        - authorName: Name of the author.
        - birthday: Birthdates of the author.
        - biography: Brief or detailed information of the author’s background, achievements, or notable information.
        - country: The country where the author is from.
        - image: Image reference or path for the author.
    - Purpose: Stores details about the author's information, including their names, images, and countries they originate from.
- Publisher Table
    - Attributes:
        - publisherId: Unique identifier for each publisher. (Primary Key)
        - publisherName: Name of the publisher.
        - address: Physical address or location details of the publisher’s headquarters or office.
    - Purpose: Serves as a repository for details such as the publisher names and location.
- Purchase Table:
    - Attributes:
        - purchaseId: Unique identifier for each Purchase (Primary Key).
        - customerId: References the id in the Customer table (Foreign Key).
        - purchaseTime: Timestamp for when the Purchase was placed.
        - deliveryTime: Timestamp for when the Purchase is expected to be delivered.
        - quantity: Number of books bought.
        - total: Total price of the books bought.
        - image: Image reference or path for the purchase bill.
    - Purpose: Records information about customer Purchases, their status, and related details such as timing and any special notes.
- Rating Table:
    - Attributes:
        - ratingId: Unique identifier for each rating. (Primary Key)

        - customerId: References the id in the Customer table (Foreign Key).
        - bookId: References the id in the Book table (Foreign Key).
        - rating: Rating given by the customer for a specific book.
    - Purpose: Records ratings provided by customers for books, allowing for feedback and potential recommendation systems based on user preferences.
- Admin Table:
    - Attributes:
        - adminId: Unique identifier for each admin user.
        - adminName: Name of the admin user.
        - pass: Password for admin login (Note: Similar to customer passwords, secure storage practices are crucial here).
        - email: Email address of the admin user.
    - Purpose: Stores information about admin users who manage the system, granting access control and administrative privileges to authorized users.

Table Relational Schema in Entity-Relationship Diagram



1. **Customer - Purchase (1:N relationship):**

   A customer can have multiple purchase transactions over time, but each purchase is made by a single customer. This relationship allows one customer (1) to be associated with multiple purchases (N).

2. **Book - Author (N:1 relationship):**

   Each book (1) is authored by one author (1), but an author (1) can have authored multiple books (N). This relationship indicates that many books can be associated with one author, creating a many-to-one relationship between books and authors.

3. **Book - Publisher (N:1 relationship):**

Each book (1) is published by one publisher (1), but a publisher (1) can publish multiple books (N). This relationship indicates that many books can be associated with one publisher, creating a many-to-one relationship between books and publishers.

4. **Book - Rating (1:N relationship):**
   A book can have many ratings from multiple users, which means one book (1) can be associated with multiple ratings (N).
5. **Customer - Rating (1:N relationship):**
   A customer (1) can give multiple ratings (N) for different books. Each rating (N) is for one specific book (1). This relationship shows that customers rate multiple books while each rating belongs to a single book.

Relational Schema

```
Customer (
        customerId INT NOT NULL AUTO_INCREMENT,
        customerName VARCHAR(55) NOT NULL,
        gender VARCHAR(15),
        pass VARCHAR(55),
        email VARCHAR(55),
        address VARCHAR(55),
        PRIMARY KEY (customerId)
);

Book (
        bookId INT NOT NULL AUTO_INCREMENT,
        title VARCHAR(55) NOT NULL,
        price INT,
        quantity INT,
        authorId INT,
        image VARCHAR(55),
        synopsis VARCHAR(200),
        publisherId INT,
```

```
        PRIMARY KEY (bookId),
        FOREIGN KEY (authorId) REFERENCES Author(authorId),
        FOREIGN KEY (publisherId) REFERENCES Publisher(publisherId)
);

Author (
        authorId INT NOT NULL AUTO_INCREMENT,
        authorName VARCHAR(55) NOT NULL,
        birthday DATETIME,
        biography VARCHAR(200),
        country VARCHAR(55),
        image VARCHAR(55),
        PRIMARY KEY (authorId)
);

Publisher (
        publisherId INT NOT NULL AUTO_INCREMENT,
        publisherName VARCHAR(55) NOT NULL,
        address VARCHAR(55),
        PRIMARY KEY (publisherId)
);

Purchase (
        purchaseId INT NOT NULL AUTO_INCREMENT,
        customerId INT,
        purchaseTime DATETIME,
        deliveryTime DATETIME,
        quantity INT,
        total INT,
        image VARCHAR(55),
        FOREIGN KEY (customerId) REFERENCES Customer(customerId),
```

```
        PRIMARY KEY (purchaseId)
);

Rating (
        ratingId INT NOT NULL AUTO_INCREMENT,
        customerId INT,
        bookId INT,
        rating INT,
        PRIMARY KEY (ratingId),
        FOREIGN KEY (customerId) REFERENCES Customer(customerId),
        FOREIGN KEY (bookId) REFERENCES Book(bookId)
);

Admin (
        adminId INT NOT NULL AUTO_INCREMENT,
        adminName VARCHAR(55) NOT NULL,
        pass VARCHAR(55),
        email VARCHAR(55),
        PRIMARY KEY (adminId)
);
```

Normalisations

Our database system is already in the 3NF because it meets the following criteria:

1. Has no duplicate data in a single row.
2. The primary key uniquely identifies a single row.
3. The entire key uniquely identifies the row.
4. Eliminate any information in the table that doesn't have a direct relationship with the primary key.
5. Has simplified the data management process.

Sample Queries

SELECT

- To get the image string path and use it to be displayed on the page from the book with id equals to 10139.
  SELECT image FROM Book WHERE bookId = 10139);
- To get the biography string path and use it to be displayed on the page from the author with id equals to 28.
  SELECT biography FROM Author WHERE authorId = 28);
- To get information about books, including book ID, title, price, quantity, synopsis, and image, along with details about the corresponding author (author ID and name) and publisher (publisher ID and name) through inner joins on the Author and Publisher tables based on their respective IDs in the Book table.
  SELECT Book.bookId, Book.title, Book.price, Book.quantity, Book.synopsis, Book.image, Author.authorId, Author.authorName, Publisher.publisherId, Publisher.publisherName
  FROM Book
  INNER JOIN Author ON Book.authorId = Author.authorId
  INNER JOIN Publisher ON Book.publisherId = Publisher.publisherId;

UPDATE

- To modify a customer's information.
  UPDATE Customer SET customerName = ?, gender = ?, pass = ?, email = ?, address = ? WHERE customerId = ?
- To modify a book's details.
  UPDATE Book SET authorId = ?, title = ?, price = ?, quantity = ?, image = ?, synopsis = ?, publisherId = ? WHERE bookId = ?

INSERT

- To add a new record to the author table

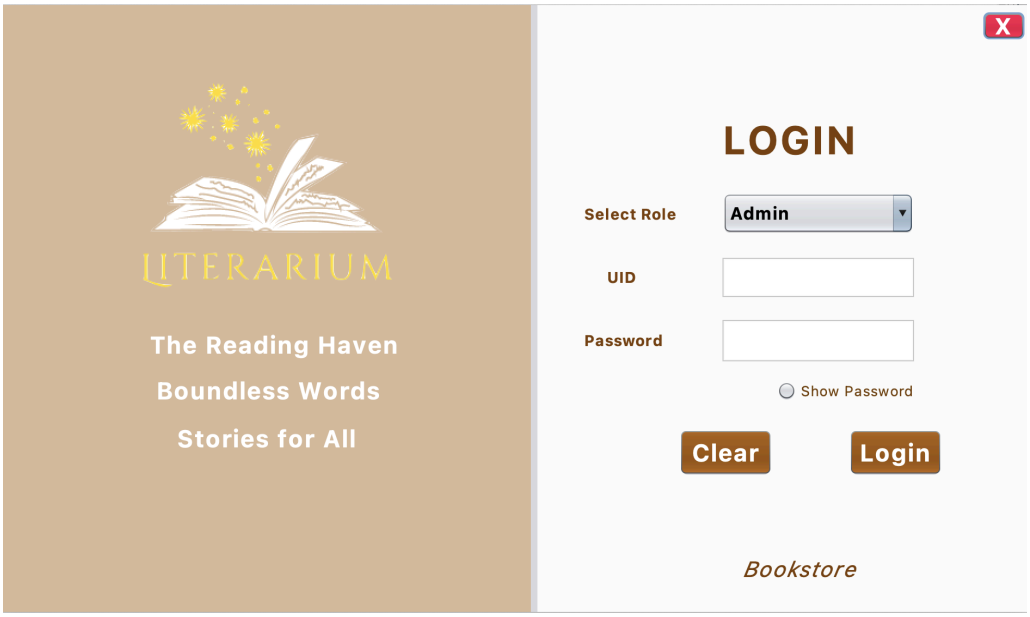
INSERT INTO Author (authorName, birthday, image, biography, country) VALUES (?, ?, ?, ?, ?)

- To add a new record to the publisher table
  INSERT INTO Publisher (publisherName, address) VALUES (?, ?)

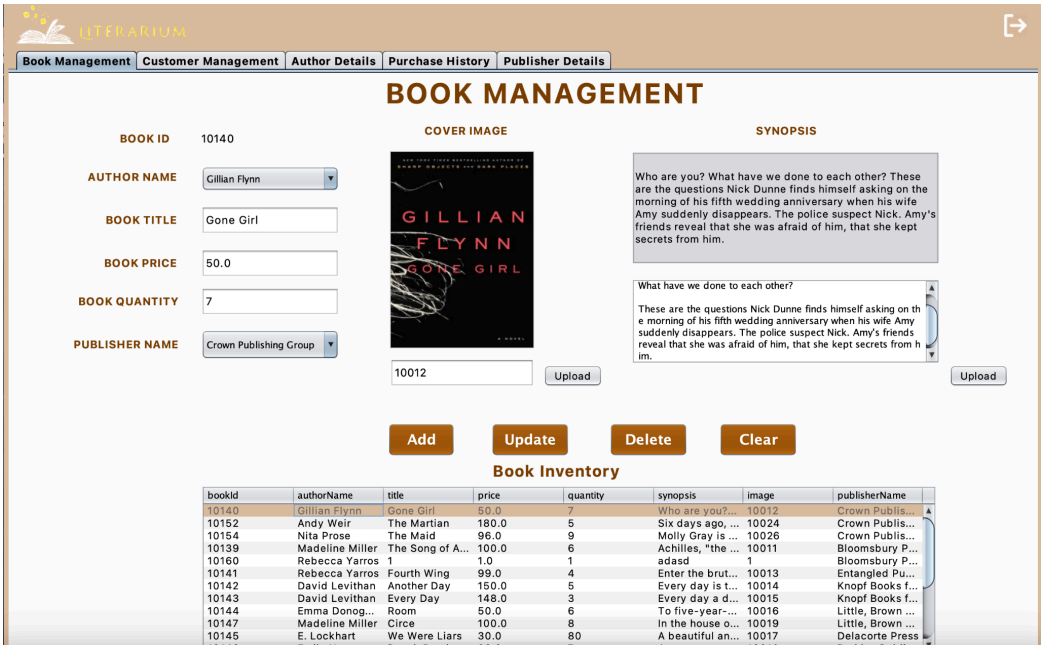
DELETE

- To remove a record from the 'Rating' table where the 'bookId' matches any 'bookId' values obtained from a subquery selecting books with a specified 'authorId' from the 'Book' table.
  DELETE FROM Rating WHERE bookId IN (SELECT bookId FROM Book WHERE authorId = ?)

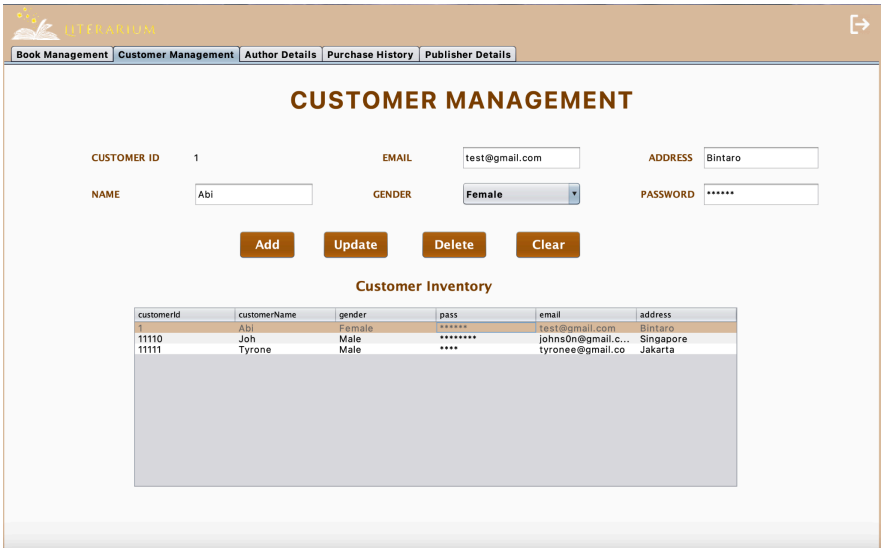
User Interfaces



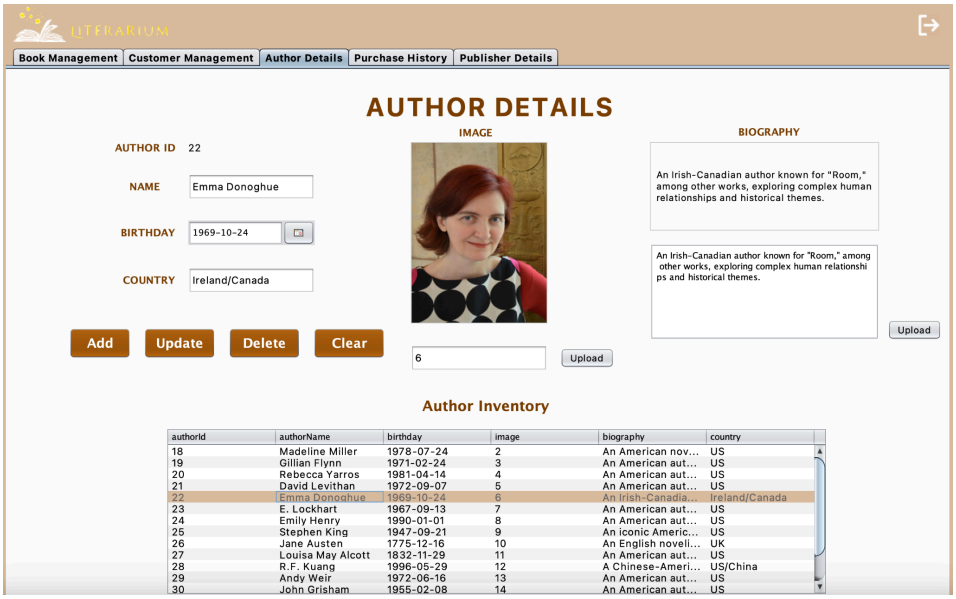
The above image shows our application's home page. Users can choose whether to log in as an admin or a customer by inserting their user ID and password.



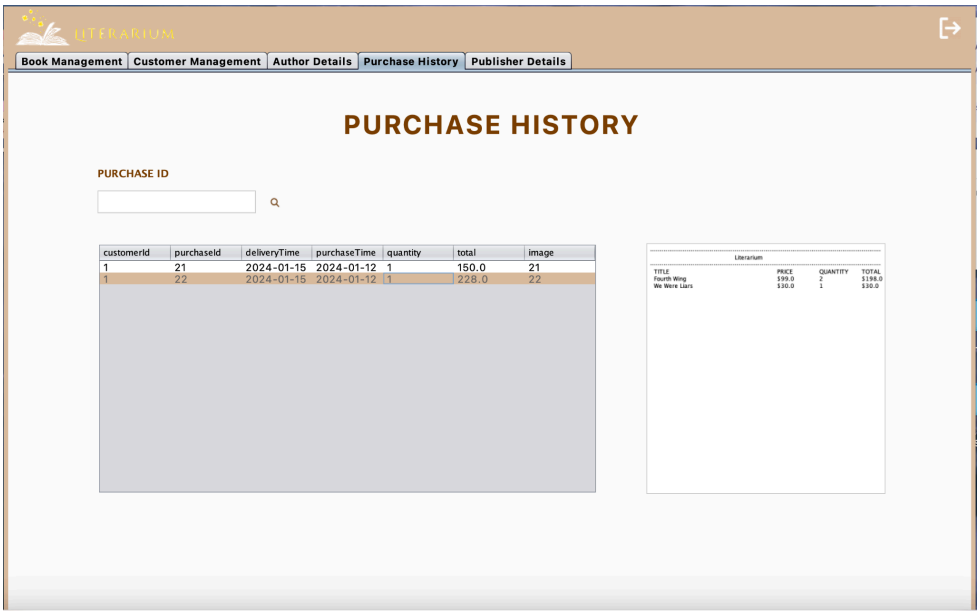
The above image shows the book management page, which helps manage book collections. It lets us (admins) add, update, delete, and track our books. It also allows us to see details like author, title, price, synopsis, etc.



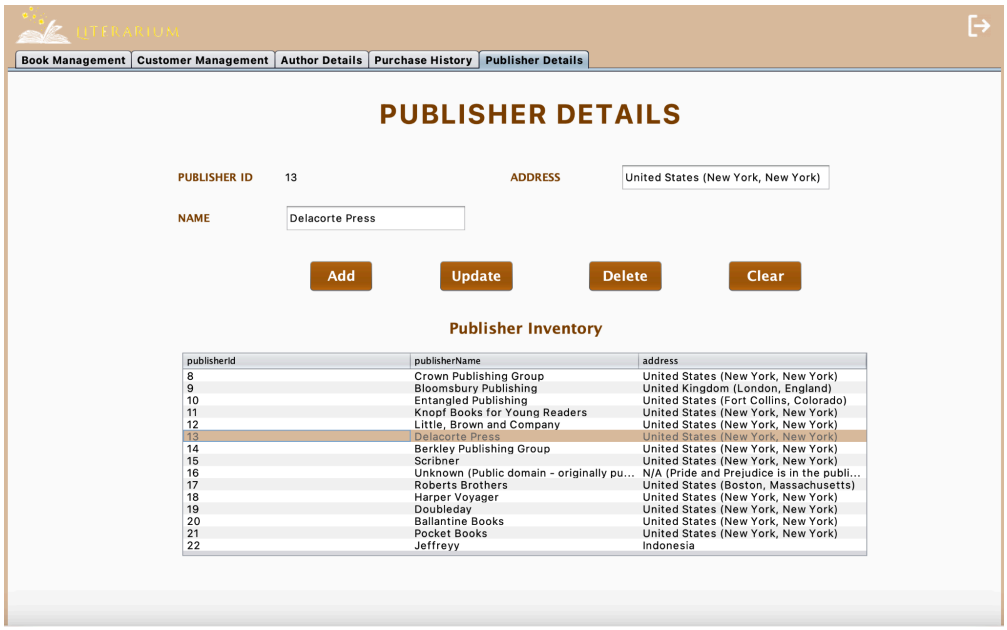
The above image shows the customer management page, which helps manage customer inventory. It lets us (admins) edit customer information and view details like name, email, and address.



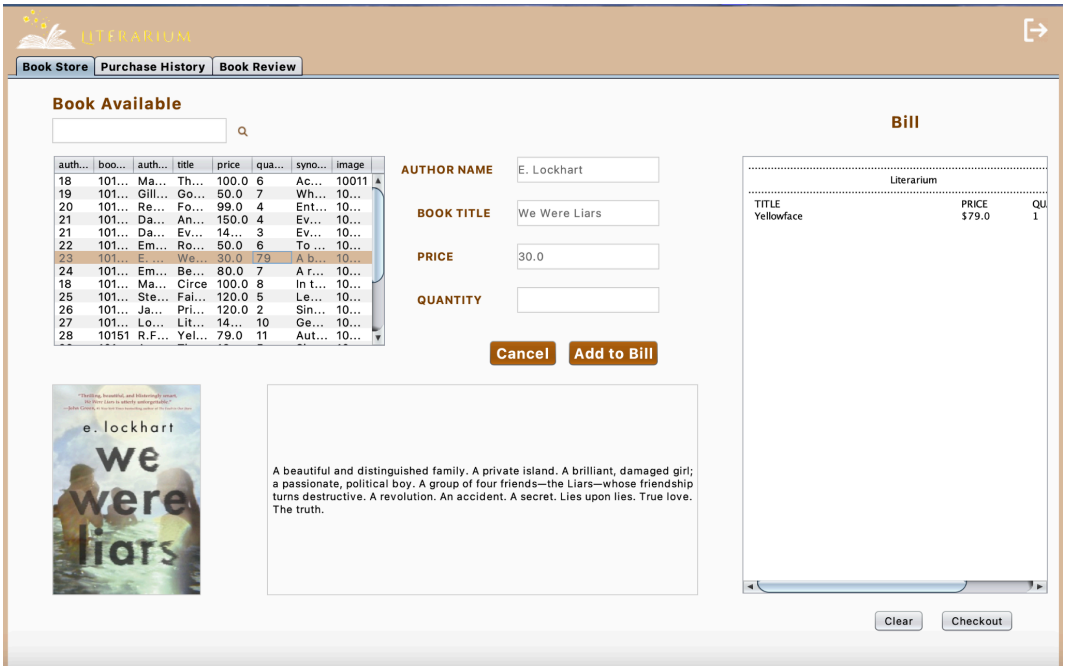
The above image shows the author details page, which helps manage the author list. It lets us (admins) add, update, delete, and track our authors. It also allows us to see details like authors' names, birthdates, biography, and country they originated from.



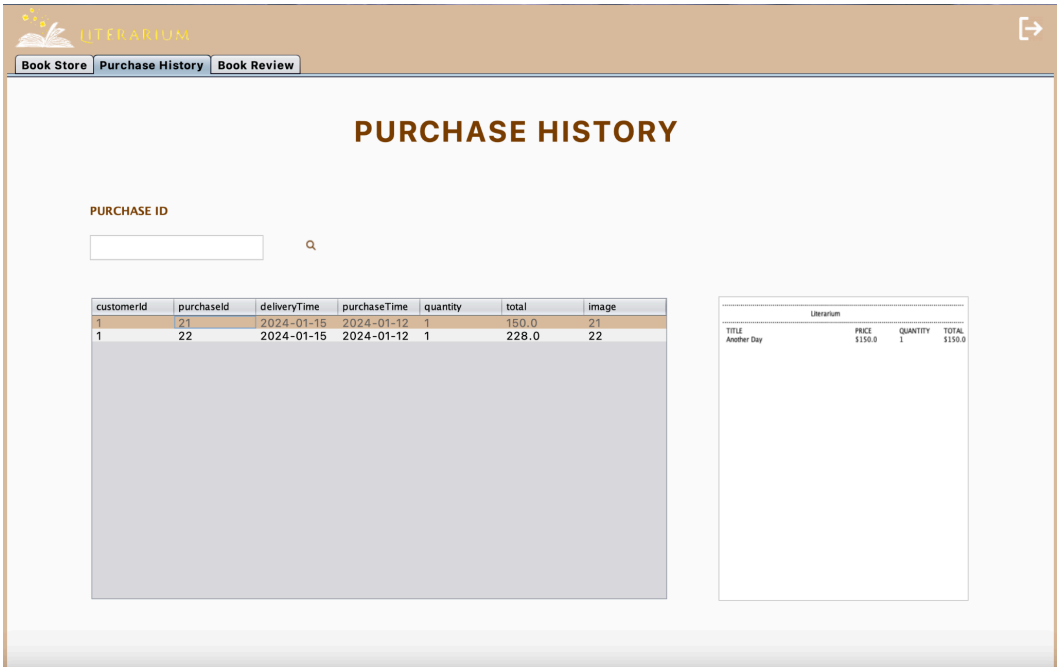
The above image shows a purchase history page. Admins can see the delivery time, purchase time, quantity, and total price of the books bought, and the image of the bill produced from all purchases made by all the customers in the bookstore.



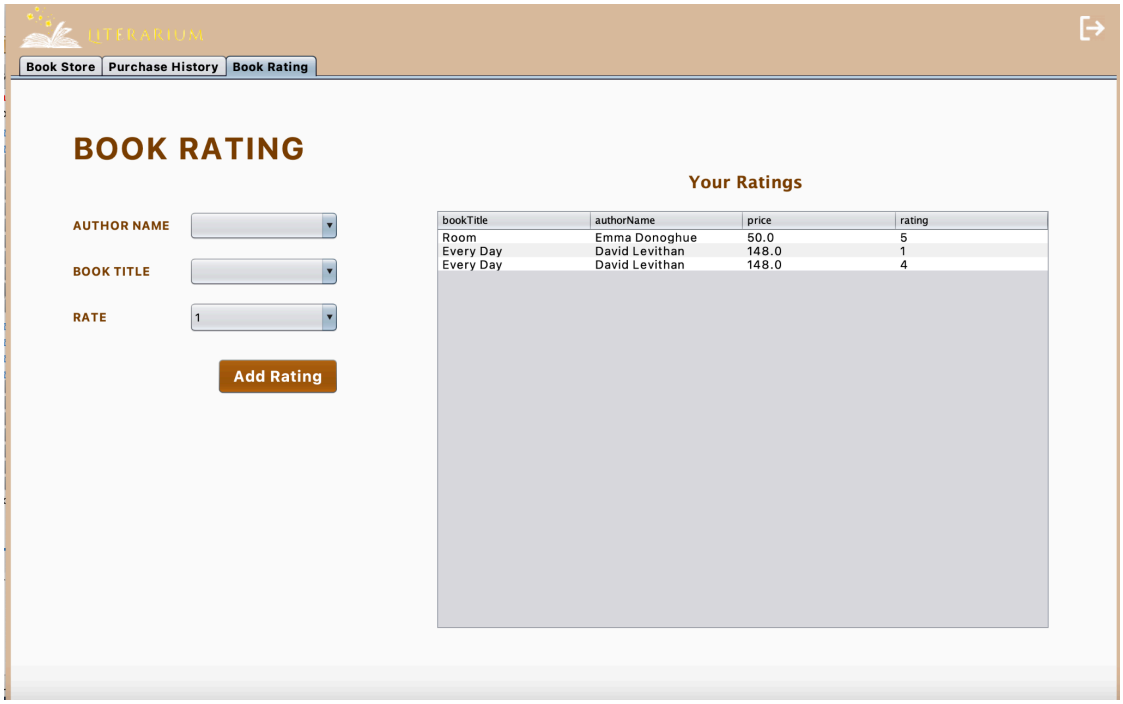
The above image shows the publisher details page, which helps manage the publisher list. It lets us (admins) add, update, delete, and track our publishers. It also allows us to see details like publishers' names and their addresses.



The above image shows the bookstore page. It allows customers to browse books, add books to a shopping cart, view shopping cart content (bill), and proceed to checkout.



The above image shows the purchase history page. It allows customers to view the purchase history that they have made and the bill image.



The above image shows the book review page. It allows customers to add a book rating and view the ratings from other customers as well.

Database Security

Our application has two roles: Admin and Customer, in which they have different database permissions as explained below:

Admin Role

- Database Permissions
    - Full access to the book, customer, author, and publisher table.
    - Permissions to execute CRUD (Create, Read, Update, and Delete) operations on those tables.
- Purchase History
    - Access to view the purchase history made by all customers.

Customer Role

- Database Permissions
    - Limited access to the book, customer, author, and publisher table.
    - Permissions to execute specific CRUD (Create, Read, Update, and Delete) operations, such as:
        - Read access to book information, including the author and publisher.
        - Create access to make purchases.
        - Read access to view their purchase history.
        - Create access to rate books.
- Purchase History
    - Access to view their purchase history.
- Ratings
    - Access to view ratings from all customers.

Appendices

Link to the GIT Website

https://github.com/audreyfabiola/Literarium-Database

Link to blog links

Abigail: https://priscillabigaill.wordpress.com/
Audrey: https://clarissaaudreyy.wixsite.com/blog
Jeffrey: https://sites.google.com/view/jeffreyyy/