DOKUZ EYLÜL UNIVERSITY

ENGINEERING FACULTY

DEPARTMENT OF COMPUTER ENGINEERING

CME 3201

DATABASE MANAGEMENT SYSTEMS

LIBRARY MANAGEMENT SYSTEM FINAL REPORT

by

Kübra Özalp 2020510085

Fatma Ceren Akyüz 2020510007

Hüveyda Başyurtlu 2020510017

Yusuf Badakol 2020510013

System Domain

FOGGY is a virtual library application. With this application, you can search for books, complete the borrowing process after making a reservation for a specific period, and also view the books you liked.

There are two types of registered users in the system: customers and administrators. Our customers can easily borrow the books they want. They can see the books they borrowed in the past and the dates on which these books were borrowed. With user-friendly interface features, our library system aims to provide readers with a more personalized book experience.

An administrator can log into the system with a registered username and password. They can view customer information, and impose necessary restrictions on their accounts.

The system should check the availability of books in the library and reservation information to prevent double borrowings, conflicts on reservation dates, and multiple reservations for the same book on the same date.

Operation list

1. Member Operations
    - Add Member
    - Update Member
    - Delete Member
2. Book Operations
    - Add Book
    - Update Book
    - Delete Book
3. Inventory Management
    - Check Inventory
    - Generate Inventory Report
4. Admin Operations
    - Add Admin
    - Update Admin
    - Delete Admin

Problems Encountered

1. Database Integration Issues: We faced difficulties in establishing a connection between the database and the project, encountering various errors. However, after thorough research, we successfully established the connection using the appropriate plugins.

2. Missing Database Entries: Initially, the lines we added to the

database were not appearing in the table. We resolved this issue by ensuring the use of correct variable names, thereby making the added entries visible in the table.

3. Errors in User Type Modifications: When making changes to user types, we encountered errors in the user bar. This was resolved by implementing the code "Session["role"] == null || Session["role"].Equals("")" to handle the errors effectively.

4. Challenges in Establishing Table Relationships: Connecting tables proved to be challenging initially. However, by using the correct primary keys, we were able to overcome this issue and successfully establish relationships between the tables.

Used technology, tools and programming languages

1. Programming Languages
    - C#: Primary programming language for developing the backend logic, business rules, and server-side functionality of the Library Management System.
    - HTML: Utilized for creating the structure and markup of the web-based user interface.
    - CSS: Responsible for styling and layout of the web pages, ensuring a visually appealing and responsive design.
    - JavaScript (JS): Used for client-side scripting to enhance user interactions and add dynamic features within the web interface.
2. Web Framework
    - ASP.NET: Web framework employed for building dynamic and scalable web applications. ASP.NET provides tools and libraries for C# development, making it suitable for creating feature-rich web solutions.
3. Database Management System
    - Microsoft SQL Server (MSSQL): Relational database management system (RDBMS) used for storing and managing data related to books, members, borrowing records, and other aspects of the Library Management System.
4. Web Development Tools:
    - Visual Studio: Microsoft's integrated development environment (IDE) utilized for C# and ASP.NET development. It provides features for coding, debugging, testing, and designing web applications.
5. Client-Side Scripting
    - JavaScript Libraries (e.g., jQuery): Libraries used to simplify and streamline client-side scripting tasks, improving the efficiency of user interface development.

ER diagram with table detail

1. Firstly a user logs in.

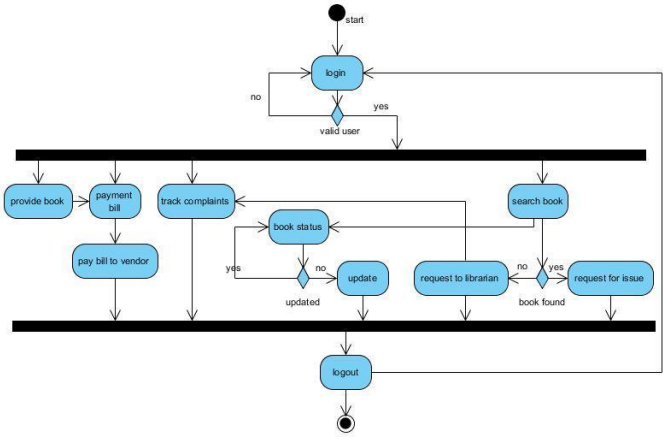


figure 1

2. If the customer requests a book.

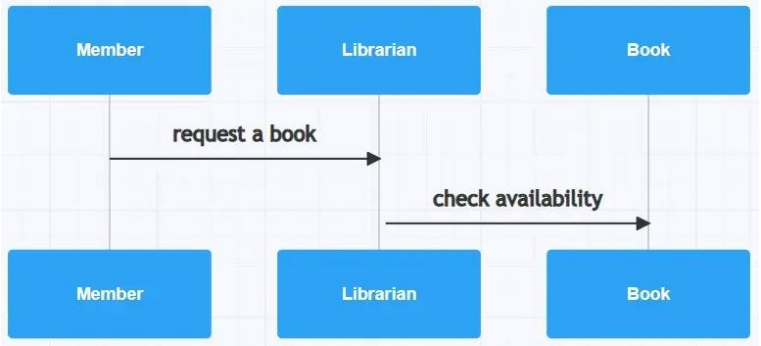


figure 2

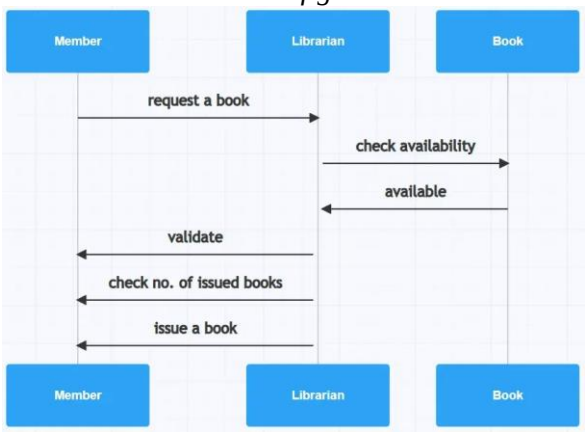


figure 3

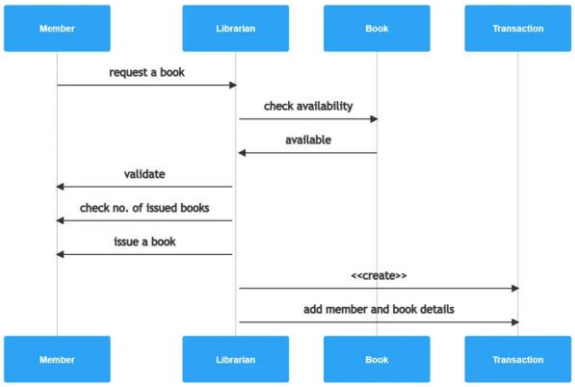


figure 4

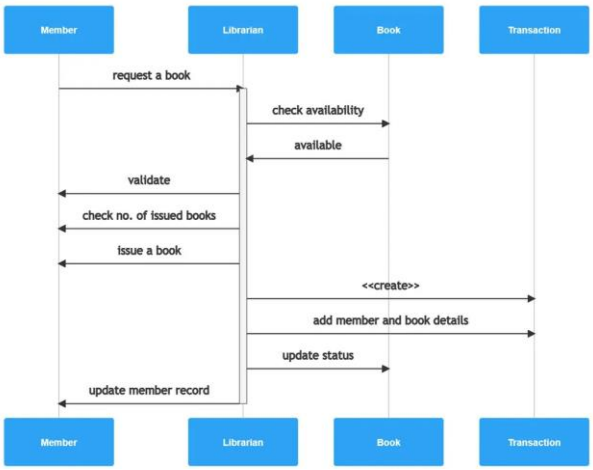


figure 5

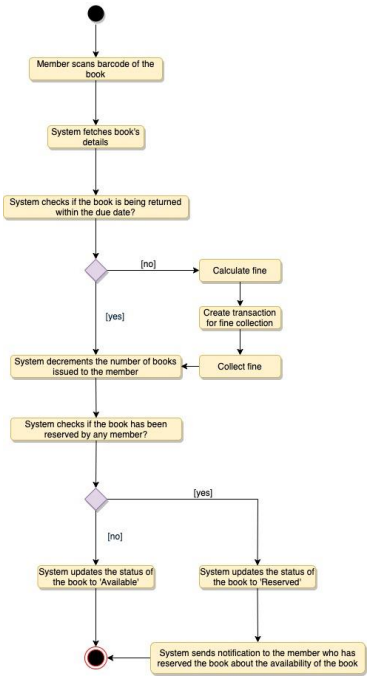


figure 6

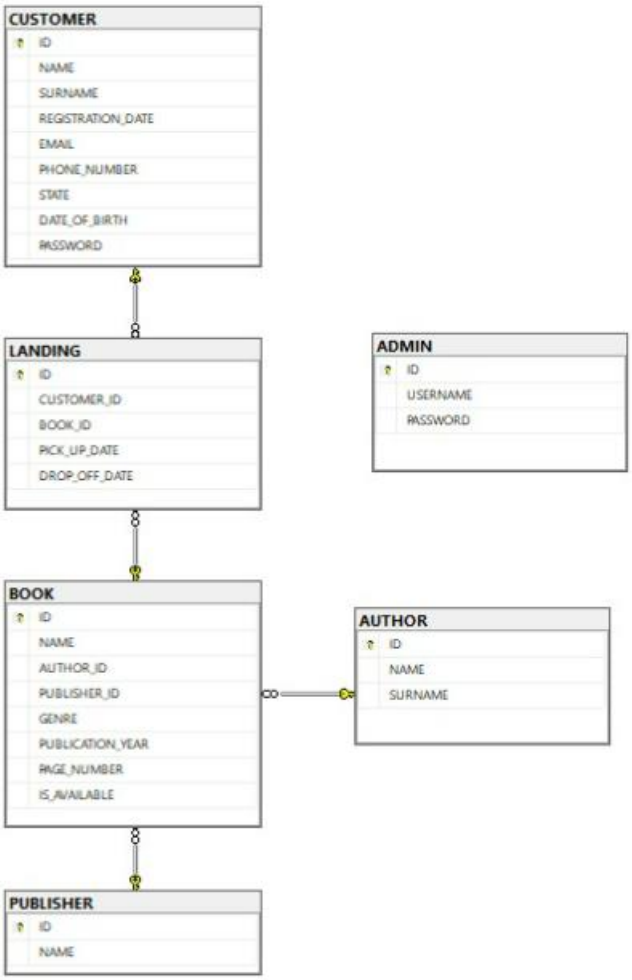


figure 7

SQL statements (basic operations on database, stored procedures, views, complex queries)

1. In adminbookinventory.aspx.cs file in deleteBookByNameAndAuthorId() function, deleting a book from Book table: `DELETE FROM BOOK WHERE NAME = @BookName AND AUTHOR_ID = @AuthorID"`

2. In adminbookinventory.aspx.cs file in updateBookByNameAndAuthorId() function, before updating a book, check if the author and the publisher of the book are exist, if not the tables, AUTHOR and PUBLISHER : `INSERT INTO PUBLISHER(NAME) VALUES(@NAME), INSERT INTO AUTHOR(NAME, SURNAME) VALUES(@NAME, @SURNAME)`

3. After that the book can be updated: `UPDATE BOOK SET NAME=@NAME, AUTHOR_ID=@AUTHOR_ID, PUBLISHER_ID=@PUBLISHER_ID, GENRE=@GENRE, PUBLICATION_YEAR=@PUBLICATION_YEAR, PAGE_NUMBER=@PAGE_NUMBER, IS_AVAILABLE=@IS_AVAILABLE WHERE NAME=@BookName AND AUTHOR_ID=@AUTHOR_ID`

4. In adminbookinventory.aspx.cs file in addNewBook() function, before inserting a book, check if the publisher and author exist, if not insert into the tables, AUTHOR and PUBLISHER : `INSERT INTO AUTHOR(NAME, SURNAME) VALUES(@NAME, @SURNAME), INSERT INTO PUBLISHER(NAME) VALUES(@NAME)`

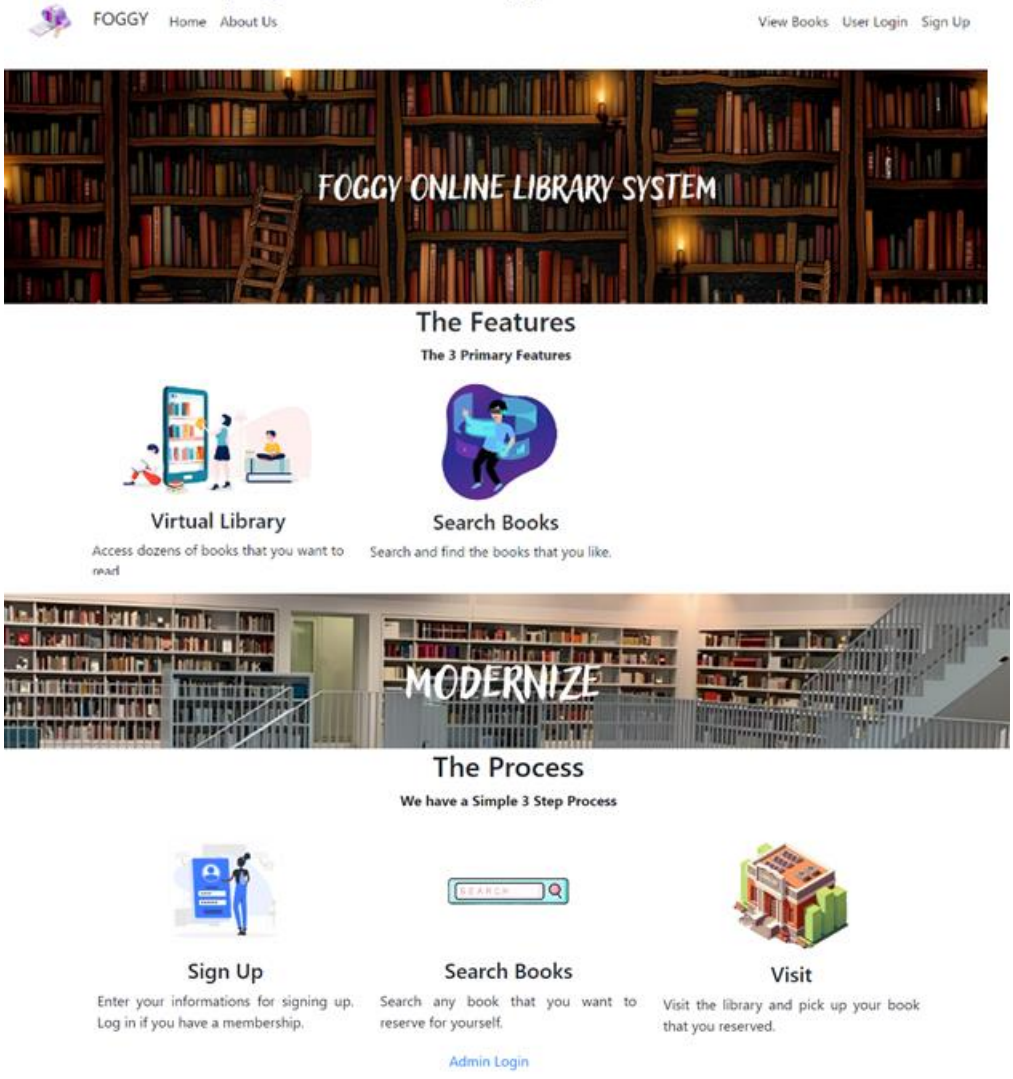
5. After that the book can be added to the BOOK table: `INSERT INTO BOOK(NAME, AUTHOR_ID, PUBLISHER_ID, GENRE, PUBLICATION_YEAR, PAGE_NUMBER, IS_AVAILABLE) VALUES(@NAME, @AUTHOR_ID, @PUBLISHER_ID, @GENRE, @PUBLICATION_YEAR, @PAGE_NUMBER, @IS_AVAILABLE)`

6. In adminlogin.aspx.cs file in Button1_Click function, which is 'Login' button, to check if the admin exist.(TextBox1 AND TextBox2 are the boxes which are the admin enter their informations): `SELECT * FROM ADMIN WHERE USERNAME='" + TextBox1.Text.Trim() + "' AND PASSWORD='" +.Text.Trim() + "'"`
7. In userlogin.aspx.cs file in Button1_Click function, which is 'Login' button, to check if the customer exist.(TextBox1 AND TextBox2 are the boxes which are the customer enter their informations): `SELECT * FROM CUSTOMER WHERE EMAIL='" + TextBox1.Text.Trim() + "' AND PASSWORD='" + TextBox2.Text.Trim() + "'"`
8. In customerbooklanding.aspx using inner join for getting the author name and surname and customer name and surname from different tables:

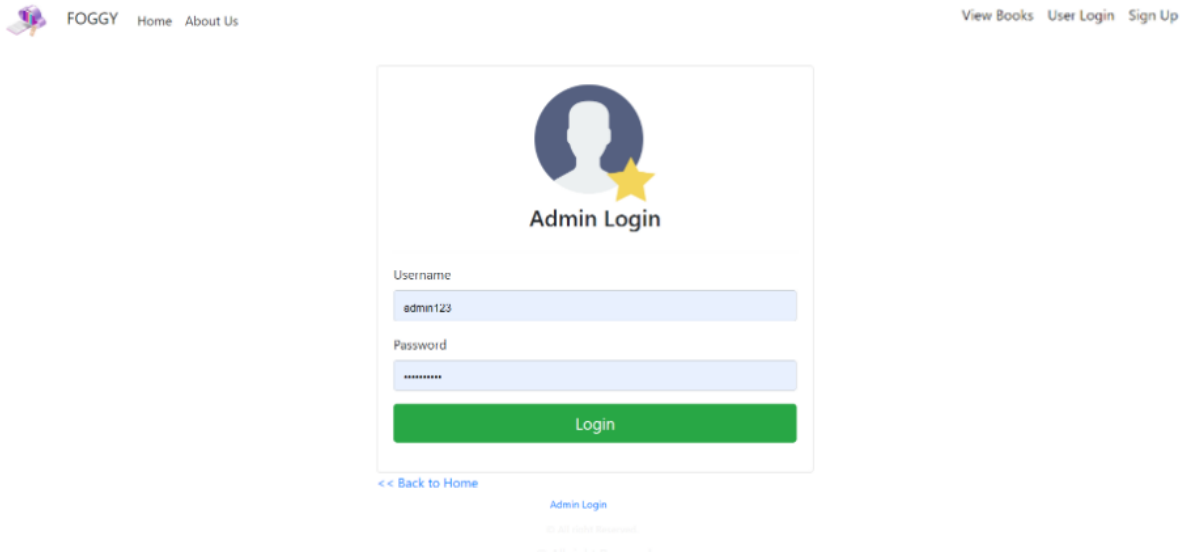
```
SELECT L.ID,C.EMAIL AS CUSTOMER_EMAIL,B.NAME S BOOK_NAME,A.NAME+''+A.SURNAME AS AUTHOR_NAME_AND_SURNAME,L.PICK_UP_DATE,L.DROP_OFF_DATE FROM LANDING L
INNER JOIN BOOK B ON L.BOOK_ID=B.ID
INNER JOIN AUTHOR A ON B.AUTHOR_ID=A.ID INNER JOIN CUSTOMER C ON L.CUSTOMER_ID=C.ID
```

Screenshots of the interfaces with a brief declaration

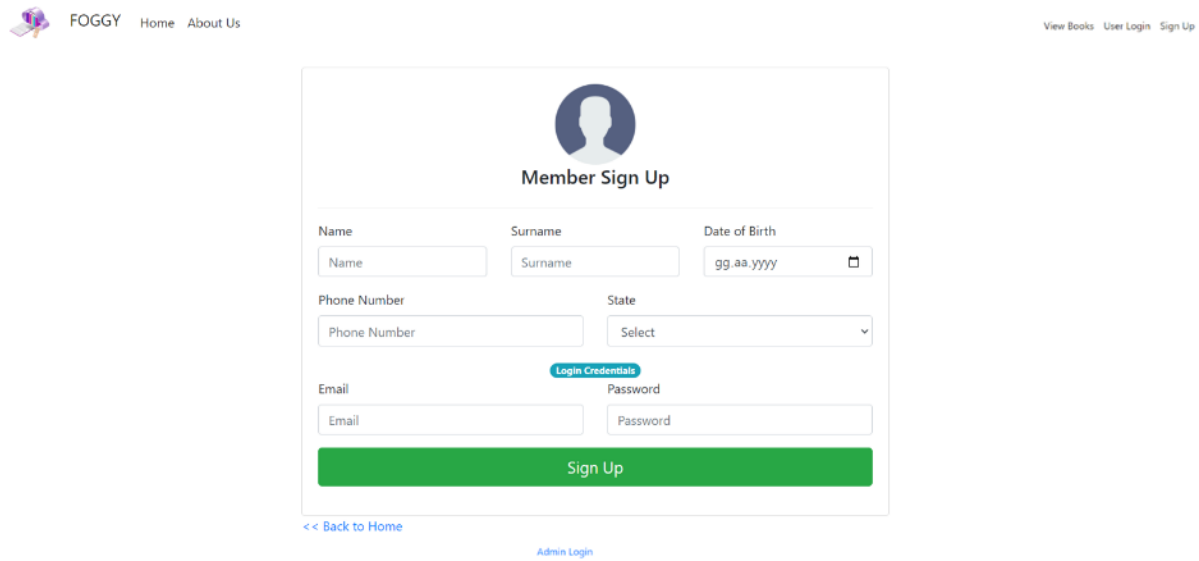
1. The front page of the web application. There are several buttons login as customer or admin, if a customer is not a member of the website they can sign up with using sign up button. Also the customer can view books. _ _



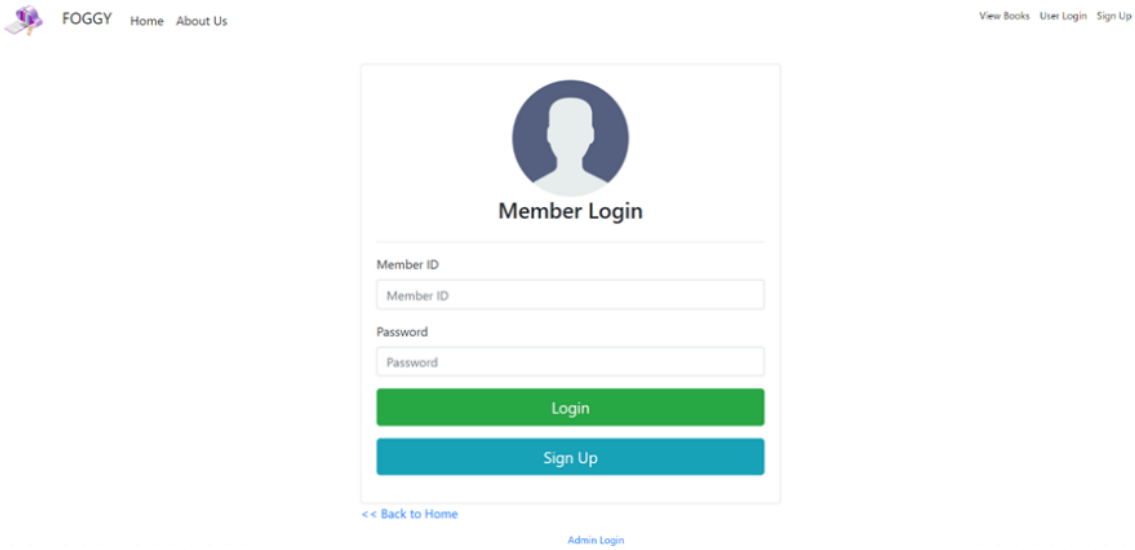
2. Admin Login Page: The admin can log in with entering their username and password.



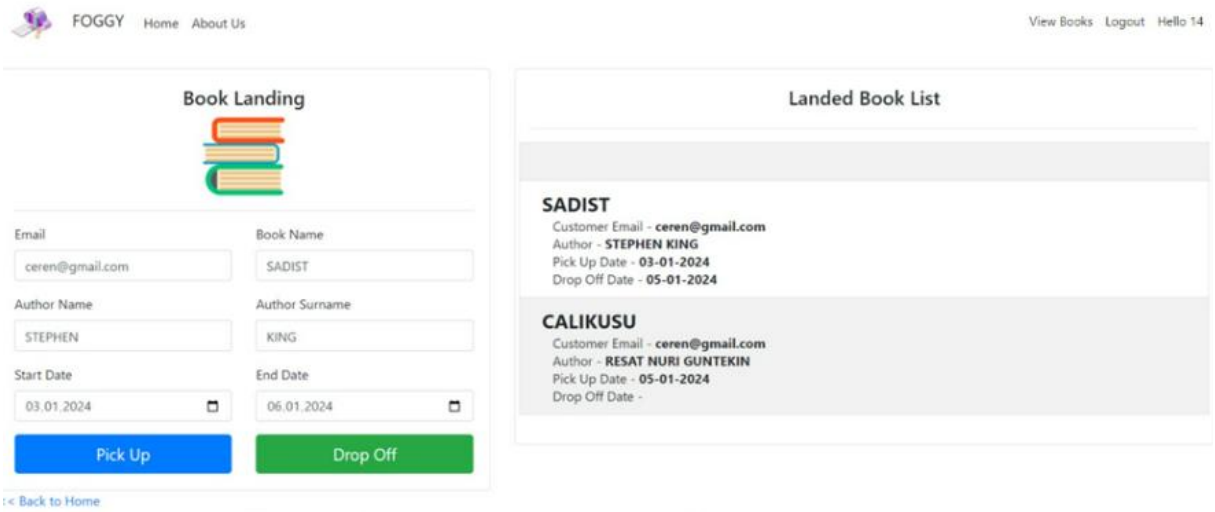
3. Member Sign Up Page: The customer can log in with entering their Name, Surname, Birthdate,Phone number, State,email and password.



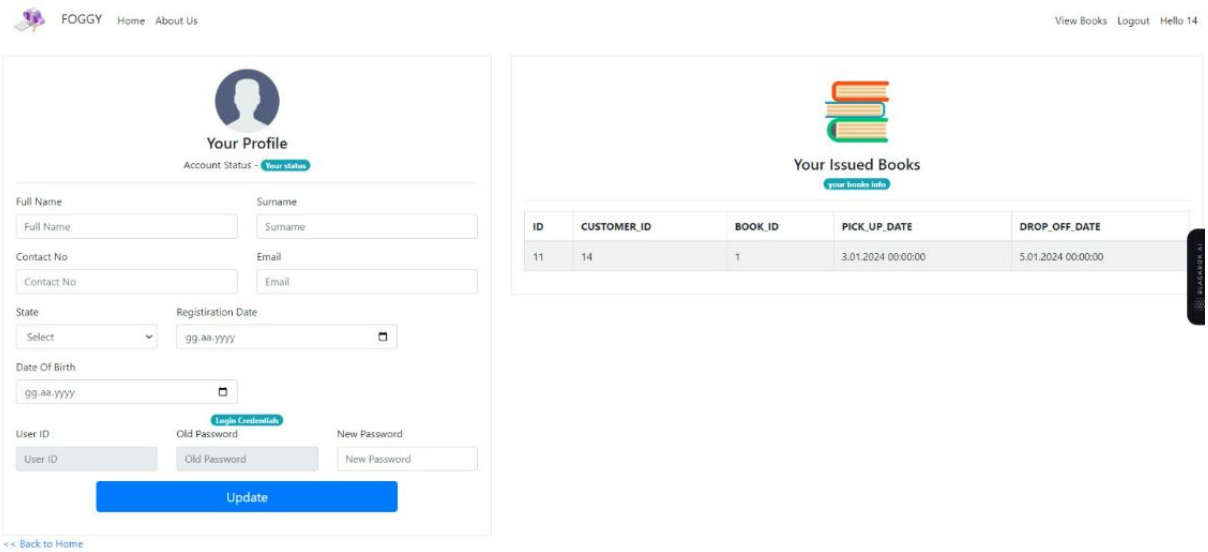
4. Customer Login Page: If the customer is the member of the library, they can log in with entering their email and password.



5. Book Landing Page: The customer can land a book with entering the book informations and clik the pick up button.



6. Customer's Profile Page: They can see the book that they landed before.



7. Customer Profile Page: The customer can see their informations.