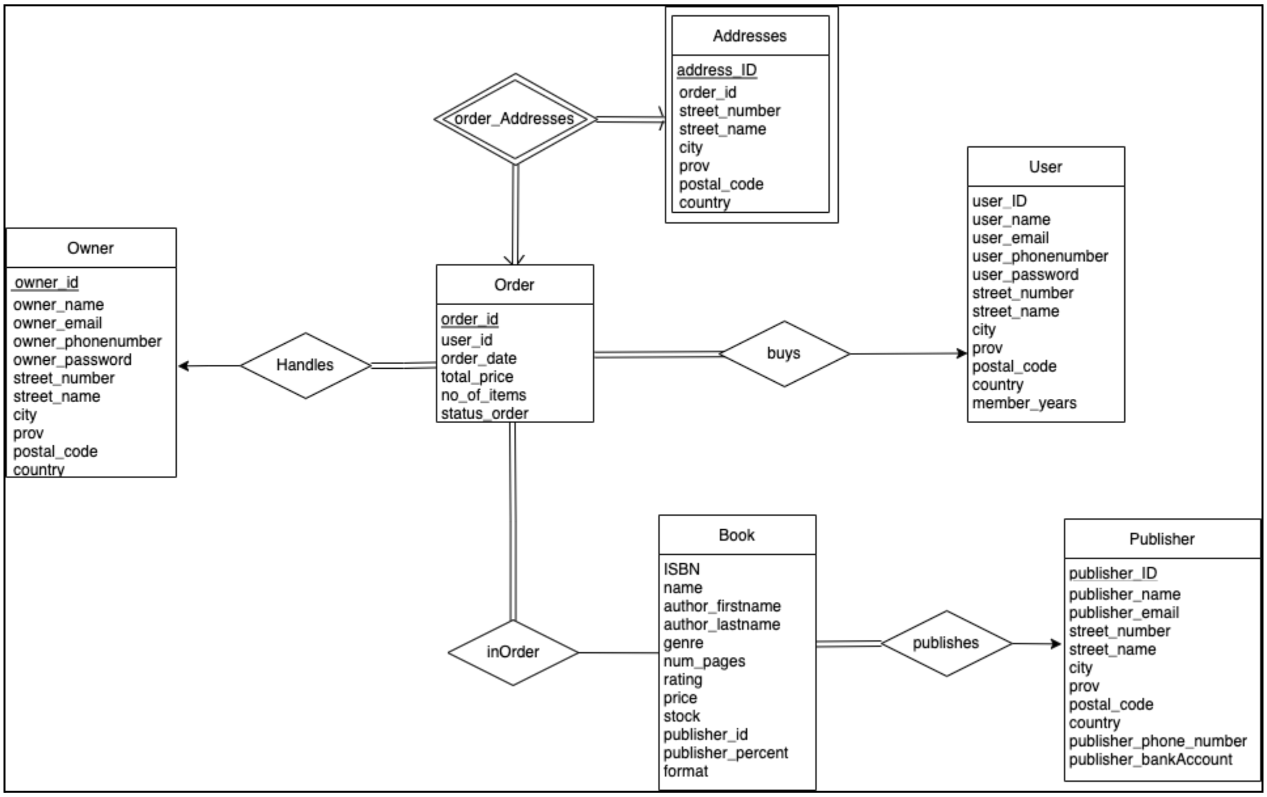
Project Report
COMP3005

Indumini Jayakody | 101109229

Samar Hussein | 101105020

ER Diagram



2.1. Conceptual Design

- *Entities*:
    - Orders
        - Represents all the books a user has bought in the one purchase
        - Uniquely identified by order_id, which also acts as the tracking number for the order
        - Price, number_items, status, date is recorded and stored
        - Status - is the current status of order (initially “Pending Shipment”)
    - Users
        - Is the online customer to the store, that has a registered account in the website
        - Uniquely identified by user_id
        - Other attributes include: Name, Email. Phone, Password, Member_years, StreetNum, StreetName, City, prov, PostalCode, Country
    - Owners
        - Is the owner and employee of the online store, also has a registered account with the website. In which they are able to view reports and handle orders
        - Uniquely identified by owner_id
        - Other attributes include: Name, Email. Phone, Password, Salary, StreetNum, StreetName, City, prov, PostalCode, Country

- Addresses
    - Is the shipping address of users in the online store, which could be different from the one a user has registered with (billing)
    - Uniquely identified by address_id, order_id
    - Other attributes include: Street_number, Street_name, City, Prov, Postal_code, Country
- Book
    - Is the product that is being sold on the Look Inna Book online store. Supplied by a publisher to the store to be sold for retail
    - Uniquely identified by ISBN
    - Other attributes include: Name, Author_firstname, Author_lastname, Genre, Num_pages, Rating (from 0-5), Price, Stock, Publisher_id, Publisher_percent (the percent of profit the publisher gets), Format (type of print)
- Publisher
    - The company that creates and distributes the books to different stores including Look Inna Book.
    - Uniquely identified by Publisher_id
    - Other attributes include: Name, Email, Street_number, Street_name, City, Province, Postal_code, Country, Phone_number, Bank_account

- *Relations*:
    - Handles
        - This represents the relationship between the owner and the order. The owner is responsible for shipping out the orders, and Handles shows that relationship. The owner uses the order_ID from order to be able to ship out the order (and update its status)
        - The attributes in this relation are owner_ID and order_ID, which are the primary keys from the Owner and Order entities (respectively). The primary key for this relation is Order_ID, since it has the “many” side of the cardinality.
        - Participation: This relation has total participation on the Order side, and partial participation on the Owner side. This is because every order must have an owner, while the owner does not have to have an order (case where there are no orders in the store).
        - Cardinality: This relation has a one to many cardinality on the Owner side, because an owner can have more than one order, while an order can only have one owner).
    - Buys
        - This represents the relationship between the User and the Order (where an Order is bought by a User).
        - The attributes in this relation are Order_ID and User_ID, which are the primary keys from Order and User (respectively). The primary key for this relation is order_ID, since it belongs to the “many” side of the cardinality.

- Participation: This relation has total participation on the order side, and partial participation on the user side. This is because an order must have a user, while a user does not need to have an order.
- Cardinality: This relation has a many to one cardinality from the order side (ie. a user can have multiple orders, while an order can only have one user)

- InOrder
    - This represents the relationship between a book and the order. Every book that is in the order is represented by this relation.
    - The attributes in this relation are order_ID and ISBN, which are the primary keys for the Order and Book entity (respectively). Both order_ID and ISBN are also the primary keys for the inOrder relation since the relationship is many-to-many.
    - Participation: This relation has total participation in the Order side, and partial participation on the Book side. This is because every order must contain a book (since you need to have a book in the checkout basket in order to be able to complete an order). The Book entity has partial participation because not every book needs to be in an order.
    - Cardinality: This relation has a many-to-many relation because an order can have many books, and a book can be in many orders (the same book ISBN
- Publishes
    - This represents the relationship between the Book and Publisher entities, where a Publisher publishes a book.
    - The attributes in this relation are ISBN and publisher_ID from the Book and Publisher entities (respectively). The primary key for this relation is ISBN, since it belongs to the “many" side of the cardinality.
    - Participation: This relation has total participation on the Book side, and partial participation on the Publishes side. This is because every book must have a publisher, while every publisher does not necessarily need to have a book.
    - Cardinality: This relation has a many-to-one (many on the Book side, and one on the Publisher side) relation because a Book can only have one publisher, and a publisher can have many books.
- orderAddresses
    - This represents the relationship between Order and Addresses entities, where an order has an address that is used for shipping (either same as the one on file, or a new one).
    - The attributes in this relation are order_id, address_id, street_number, street_name, city, prov, postal_code, country, where order_id, address_id are the primary keys. This is because Addresses is a weak entity, making the orderAddresses relation have the attributes from Addresses.
    - Participation: This relation has total participation on both sides, because an Order must have an address, and each tuple in the Addresses entity

must be tied to an Order (with order_ID). It is also a weak entity, so it relies on an order.
    - Cardinality: This relation has a one-to-one relationship because an Order must have one address, and an address can be associated with one order (since a unique addressID is generated each time).

Assumptions:

- There is only one owner for the store (one employee)
- Both User and Owner express a form of generalization extending from the Person schema
- Owner already has an account from the creation of the online store and does not need to create any additional accounts
- Our online store only supports users/shipments to Canada and United States

2.2. Reduction to Relation Schemas

- *Users*(user_id, user_name, user_email, user_phonenumber, user_password, street_number, street_name, city, prov, postal_code, country, member_years)

- *Owners*(owner_id, owner_name, owner_email, owner_phonenumber, owner_password, street_number, street_name, city, prov, postal_code, country, salary)

- *Book*(ISBN, name, author_firstname, author_lastname, genre, num_pages, rating, price, stock, publisher_id, publisher_percent, format)

- *Orders*(order_id, user_id, order_date, total_price, no_of_items, status_order)

- *Publisher*(publisher_id, publisher_name, publisher_email, street_number, street_name, city, prov, postal_code, country, publisher_phone_number, publisher_bankAccount)

- *Publishes*(ISBN, publisher_id)

- *InOrder*(order_id, ISBN)

- *Buys*(user_id, order_id)

- *Handles*(order_id, owner_id)

- *Addresses*(order_id, address_id, street_number, street_name, city, prov, postal_code, country)

- *Order_addresses*(order_id, address_id, street_number, street_name, city, prov, postal_code, country) - **Weak entity relationship is redundant here**

2.3. Normalization of Relation Schemas

Users(user_id, user_name, user_email, user_phonenumber, user_password, street_number, street_name, city, prov, postal_code, country, member_years)

Functional Dependencies:

user_id → user_name, user_email, user_phonenumber, user_password, street_number, street_name, city, prov, postal_code, country, member_years

user_email → user_id

user_email → user_password

user_email → member_years

postal_code → city, prov, country

prov → country

Explanation:

- *User_id is the primary key for users schema therefore it uniquely implies all the other attributes.* ***{user_id}$^+$ = user_id, user_name, user_email, user_phonenumber, user_password, street_number, street_name, city, prov, postal_code, country, member_years*** *therefore it is in BCNF with user_id being a superkey*
- *{user_email}$^+$ =* ***user_id, user_name, user_email, user_phonenumber, user_password, street_number, street_name, city, prov, postal_code, country, member_years*** *therefore it is also in BCNF*
- *But users schema is not in BCNF due to the violating dependencies postal_code → city, prov, country and prov → country*
- *Decomposition of users = R_1(user_id, user_name, user_email, user_phonenumber, user_password, street_number, street_name, postal_code, member_years) and R_2(postal_code, city, prov, country)*
- *Which is a lossless decomposition since R_1 U R_2 = users relation*

Owners(owner_id, owner_name, owner_email, owner_phonenumber, owner_password, street_number, street_name, city, prov, postal_code, country, salary)

Functional Dependencies:

owner_id → owner_name, owner_email, owner_phonenumber, owner_password, street_number, street_name, city, prov, postal_code, country, salary owner_email → owner_id, owner_name, owner_phonenumber, owner_password, street_number, street_name, city, prov, postal_code, country, salary

salary → owner_id, owner_name, owner_email, owner_phonenumber, owner_password, street_number, street_name, city, prov, postal_code, country

postal_code → city, prov, country

prov → country

Explanation:

- owner_id *is the primary key for owners schema therefore it uniquely implies all the other attributes.* ***{*****owner_id*****}***$^{+}$ **= owner_id*****,*** **owner_name, owner_email, owner_phonenumber, owner_password, street_number, street_name, city, prov, postal_code, country, salary**
    - *Therefore it is in BCNF with* owner_id *being a superkey*
- ***{*****owner_email*****}***$^{+}$ **= owner_id*****,*** **owner_name, owner_email, owner_phonenumber, owner_password, street_number, street_name, city, prov, postal_code, country, salary**
    - *Therefore it is also in BCNF*
- *But owners schema is not in BCNF due to the violating dependencies postal_code → city, prov, country and prov → country*
- *Decomposition of users = R_1(owner_id, user_name, user_email, user_phonenumber, user_password, street_number, street_name, postal_code, salary) and R_2(postal_code, city, prov, country)*
- *Which is a lossless decomposition since R_1 U R_2 = owners relation*

Book(ISBN, name, author_firstname, author_lastname, genre, num_pages, rating, price, stock, publisher_id, publisher_percent, format)

Functional Dependencies:

ISBN → name, author_firstname, author_lastname, genre, num_pages, rating, price, stock, publisher_id, publisher_percent, format

name, author_firstname, author_lastname, genre, price, format → ISBN

Explanation:

- *ISBN is the primary key for books schema therefore it uniquely implies all the other attributes.* ***{ISBN}***$^{+}$ ***= ISBN,*** **name, author_firstname, author_lastname, genre, num_pages, rating, price, stock, publisher_id, publisher_percent, format** *therefore it is in BCNF with ISBN being a superkey*

- **{name, author_firstname, author_lastname, genre, price, format }**$^+$ = ***ISBN,*** **name, author_firstname, author_lastname, genre, num_pages, rating, price, stock, publisher_id, publisher_percent, format** *therefore it is also in BCNF*
- *Therefore the relation is in BCNF*

Orders(order_id, user_id, order_date, total_price, no_of_items, status_order)

Functional Dependencies:

order_id → user_id, order_date, total_price, no_of_items, status_order

Explanation:

- order_id *is the primary key for orders schema therefore it uniquely implies all the other attributes.* **{order_id}**$^+$ **= user_id, order_date, total_price, no_of_items, status_order** *therefore it is in BCNF with* order_id *being a superkey*
- *Therefore the relation is in BCNF*

Publisher(publisher_id, publisher_name, publisher_email, street_number, street_name, city, prov, postal_code, country, publisher_phone_number, publisher_bankAccount)

Functional Dependencies:

publisher_id → publisher_name, publisher_email, street_number, street_name, city, prov, postal_code, country, publisher_phone_number, publisher_bankAccount

publisher_name → publisher_id, publisher_email, street_number, street_name, city, prov, postal_code, country, publisher_phone_number, publisher_bankAccount

publisher_email → publisher_id, publisher_name, street_number, street_name, city, prov, postal_code, country, publisher_phone_number, publisher_bankAccount

publisher_bankAccount → publisher_id, publisher_name, publisher_email, street_number, street_name, city, prov, postal_code, country, publisher_phone_number

postal_code → city, prov, country

prov → country

Explanation:

- publisher_id *is the primary key for publisher schema therefore it uniquely implies all the other attributes.* **{publisher_id}$^+$ = publisher_id, publisher_name, publisher_email, street_number, street_name, city, prov, postal_code, country, publisher_phone_number, publisher_bankAccount**
    - *Therefore it is in BCNF with* owner_id *being a superkey*
- **{publisher_name}$^+$ = publisher_id, publisher_name, publisher_email, street_number, street_name, city, prov, postal_code, country, publisher_phone_number, publisher_bankAccount**
    - *Therefore it is also in BCNF*
- **{publisher_email}$^+$ = publisher_id, publisher_name, publisher_email, street_number, street_name, city, prov, postal_code, country, publisher_phone_number, publisher_bankAccount**
    - *Therefore it is also in BCNF*
- **{publisher_bankAccount}$^+$ = publisher_id, publisher_name, publisher_email, street_number, street_name, city, prov, postal_code, country, publisher_phone_number, publisher_bankAccount**
    - *Therefore it is also in BCNF*
- *But publishers schema is not in BCNF due to the violating dependencies postal_code → city, prov, country and prov → country*
- *Decomposition of users = R_1(publisher_id, publisher_name, publisher_email, street_number, street_name, postal_code, publisher_phone_number, publisher_bankAccount) and R_2(postal_code, city, prov, country)*
- *Which is a lossless decomposition since R_1 U R_2 = publisher relation*

Publishes(ISBN, publisher_id)

Functional Dependencies:

ISBN → publisher_id

Explanation:

- ISBN *is the primary key for publishes schema therefore it uniquely implies all the other attributes.* **{ISBN}$^+$ = ISBN, publisher_id**
    - *Therefore it is in BCNF*

InOrder(order_id, ISBN)

Functional Dependencies:

______order_id, ISBN → ISBN

order_id, ISBN → order_id

Explanation:

- Order_id and ISBN *are the primary key for inOrder schema and the dependencies are also trivial therefore the relation is in BCNF*

Buys(user_id, order_id)

Functional Dependencies:

order_id → user_id

Explanation:

- order_id *is the primary key for buys schema therefore it uniquely implies all the other attributes.* **{**order_id**}**$^{+}$ **= user_id, order_id** *therefore it is in BCNF with* order_id *being a superkey*
- *Therefore the relation is in BCNF*

Handles(order_id, owner_id)

Functional Dependencies:

order_id → owner_id

Explanation:

- order_id *is the primary key for handles schema therefore it uniquely implies all the other attributes.* **{order_id}**$^{+}$ **= owner_id, order_id** *therefore it is in BCNF with* order_id *being a superkey*
- *Therefore the relation is in BCNF*

Addresses(order_id, address_id, street_number, street_name, city, prov, postal_code, country)

Functional Dependencies:

address_id → order_id, street_number, street_name, city, prov, postal_code, country

order_id → address_id, street_number, street_name, city, prov, postal_code, country postal_code → city, prov, country

prov → country

- *Order_id and address_id are the primary key for Addresses schema therefore it uniquely implies all the other attributes.* ***{address_id}***$^{+}$ **= address_id, order_id, street_number, street_name, city, prov, postal_code, country**
    - *Therefore it is in BCNF*
- ***{order_id}***$^{+}$ **= address_id, order_id, street_number, street_name, city, prov, postal_code, country**
    - *Therefore it is also in BCNF*
- *But addresses schema is not in BCNF due to the violating dependencies postal_code → city, prov, country and prov → country*
- *Decomposition of users =* R_1*(address_id, order_id, street_number, street_name, postal_code) and* R_2*(postal_code, city, prov, country)*
- *Which is a lossless decomposition since* R_1 *U* R_2 *= Addresses relation*

Order_addresses(order_id, address_id, street_number, street_name, city, prov, postal_code, country)

Functional Dependencies:

address_id → order_id, street_number, street_name, city, prov, postal_code, country

order_id → address_id, street_number, street_name, city, prov, postal_code, country

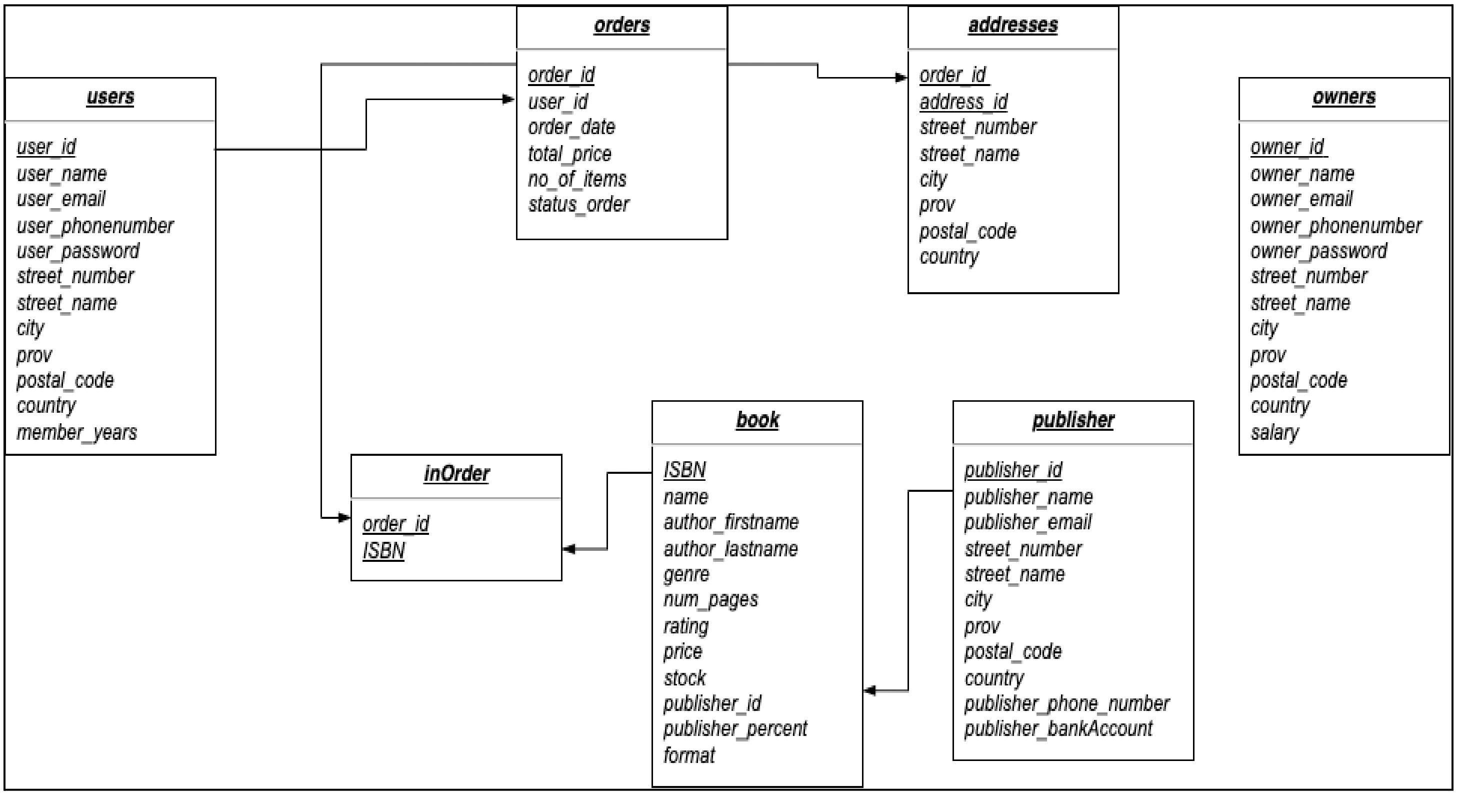
postal_code → city, prov, country

prov → country

Explanation:

- This relation has the same schema as Addresses (previous), since Addresses is a weak entity.

2.4. Database Schema Diagram

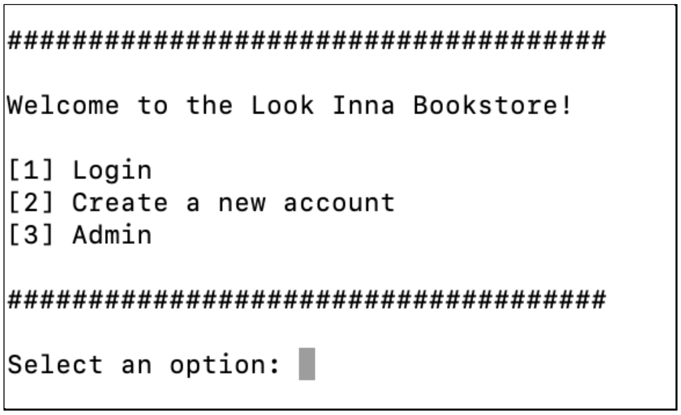


2.5. Implementation

Since we have many searches and reports, we added screenshots of a few of them for example. All functionality for the rest works properly.

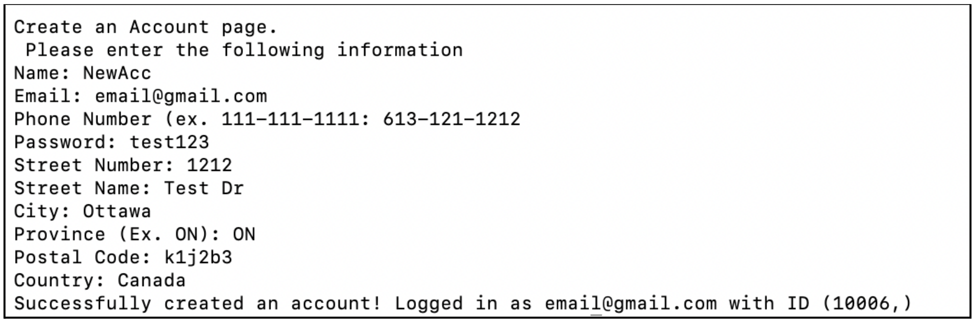
User's Interface:

- *Main Landing Page*
    - When program begins, the program user can choose to login as a User, Admin or create a new account



```
#####################################

Welcome to the Look Inna Bookstore!

[1] Login
[2] Create a new account
[3] Admin

#####################################

Select an option: █
```

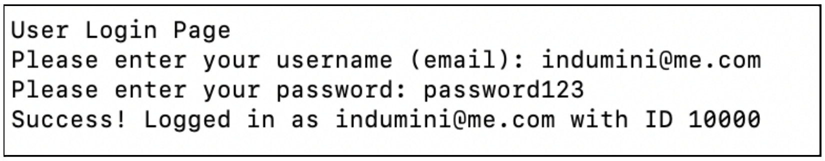
- Create new account



```
Create an Account page.
 Please enter the following information
Name: NewAcc
Email: email@gmail.com
Phone Number (ex. 111-111-1111: 613-121-1212
Password: test123
Street Number: 1212
Street Name: Test Dr
City: Ottawa
Province (Ex. ON): ON
Postal Code: k1j2b3
Country: Canada
Successfully created an account! Logged in as email@gmail.com with ID (10006,)
```

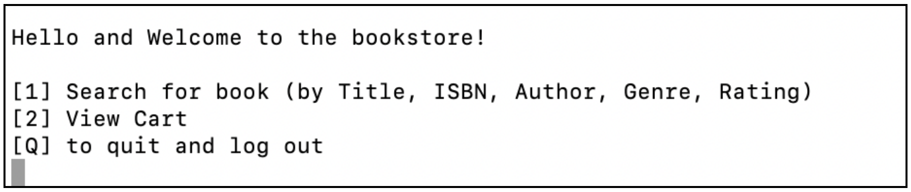
    -

- User Login
    - The user enters their username (email) and password. The input is then checked with the database and their unique userID is returned

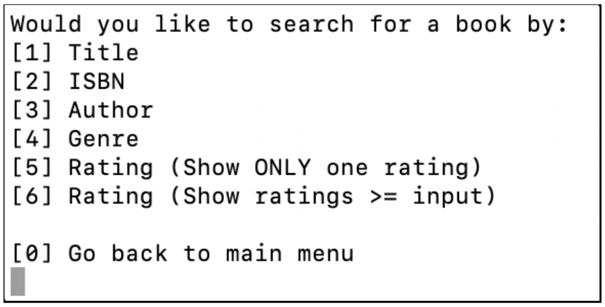


```
User Login Page
Please enter your username (email): indumini@me.com
Please enter your password: password123
Success! Logged in as indumini@me.com with ID 10000
```

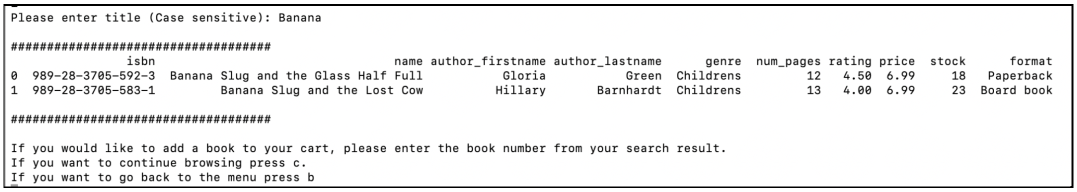
- Book Search - Title
    - After logging in, they are sent to the menu where they can choose to search for a book, view their cart or log out

    - 

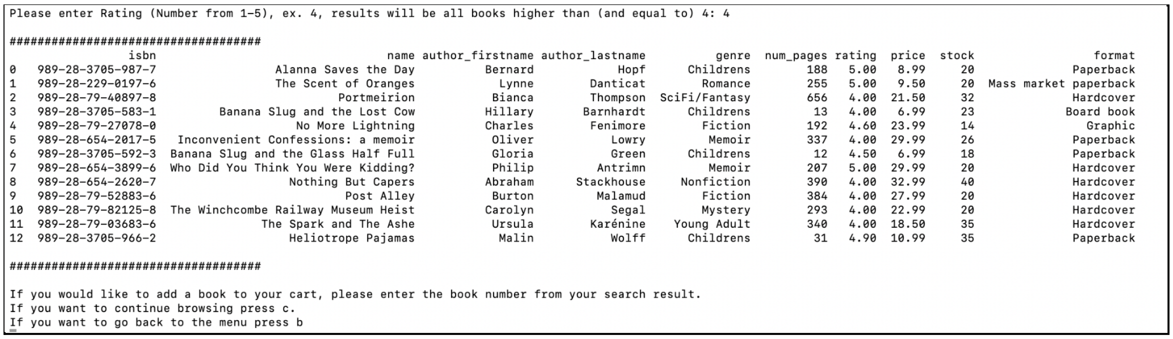
      ```
      Hello and Welcome to the bookstore!

      [1] Search for book (by Title, ISBN, Author, Genre, Rating)
      [2] View Cart
      [Q] to quit and log out
      ```

    - If the user chooses to Search (option 1), they can choose if they want to search by title, isbn, author, genre or rating
    - 

      ```
      Would you like to search for a book by:
      [1] Title
      [2] ISBN
      [3] Author
      [4] Genre
      [5] Rating (Show ONLY one rating)
      [6] Rating (Show ratings >= input)

      [0] Go back to main menu
      ```

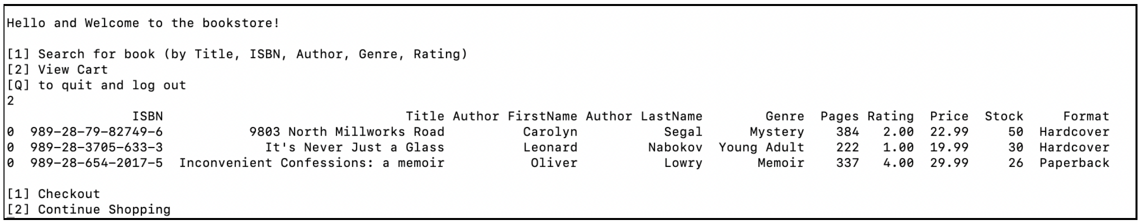
    - To search by title, the user selects 1, then they continue the search by title page, where they can enter a book title, or a keyword in a title. The search results are then printed out.
    - 

      ```
      Please enter title (Case sensitive): Banana

      ####################################
                      isbn                                 name author_firstname author_lastname      genre  num_pages rating price  stock      format
      0  989-28-3705-592-3  Banana Slug and the Glass Half Full           Gloria           Green  Childrens         12   4.50  6.99     18   Paperback
      1  989-28-3705-583-1         Banana Slug and the Lost Cow          Hillary       Barnhardt  Childrens         13   4.00  6.99     23  Board book

      ####################################

      If you would like to add a book to your cart, please enter the book number from your search result.
      If you want to continue browsing press c.
      If you want to go back to the menu press b
      ```

- Book Search - Rating
    - Similar to the Title search, the user can search by rating (inclusive) and/or search by ratings higher than input. The image below shows the results for all books with a rating of 4 and higher.
    - 

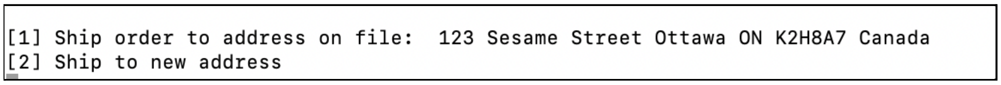
      ```
      Please enter Rating (Number from 1-5), ex. 4, results will be all books higher than (and equal to) 4: 4

      ####################################
                       isbn                                name author_firstname author_lastname        genre  num_pages rating  price  stock                 format
      0   989-28-3705-987-7                 Alanna Saves the Day          Bernard            Hopf    Childrens        188   5.00   8.99     20              Paperback
      1   989-28-229-0197-6                The Scent of Oranges            Lynne        Danticat      Romance        255   5.00   9.50     20  Mass market paperback
      2   989-28-79-40897-8                          Portmeirion           Bianca        Thompson  SciFi/Fantasy      656   4.00  21.50     32              Hardcover
      3   989-28-3705-583-1        Banana Slug and the Lost Cow          Hillary       Barnhardt    Childrens         13   4.00   6.99     23             Board book
      4   989-28-79-27078-0                 No More Lightning          Charles        Fenimore      Fiction        192   4.60  23.99     14                Graphic
      5   989-28-654-2017-5   Inconvenient Confessions: a memoir           Oliver           Lowry       Memoir        337   4.00  29.99     26              Paperback
      6   989-28-3705-592-3  Banana Slug and the Glass Half Full           Gloria           Green    Childrens         12   4.50   6.99     18              Paperback
      7   989-28-654-3899-6  Who Did You Think You Were Kidding?           Philip         Antrimn       Memoir        207   5.00  29.99     20              Hardcover
      8   989-28-654-2620-7                 Nothing But Capers          Abraham      Stackhouse   Nonfiction        390   4.00  32.99     40              Hardcover
      9   989-28-79-52883-6                          Post Alley           Burton         Malamud      Fiction        384   4.00  27.99     20              Hardcover
      10  989-28-79-82125-8  The Winchcombe Railway Museum Heist          Carolyn           Segal      Mystery        293   4.00  22.99     20              Hardcover
      11  989-28-79-03683-6                 The Spark and The Ashe           Ursula        Karénine  Young Adult        340   4.00  18.50     35              Hardcover
      12  989-28-3705-966-2                  Heliotrope Pajamas            Malin           Wolff    Childrens         31   4.90  10.99     35              Paperback

      ####################################

      If you would like to add a book to your cart, please enter the book number from your search result.
      If you want to continue browsing press c.
      If you want to go back to the menu press b
      ```

- View Cart
    - Once a book has been added to the cart (by typing the book number from the search), the cart can be viewed from the menu. When clicked, the user's cart can be seen. Here the user can choose to continue shopping or checkout



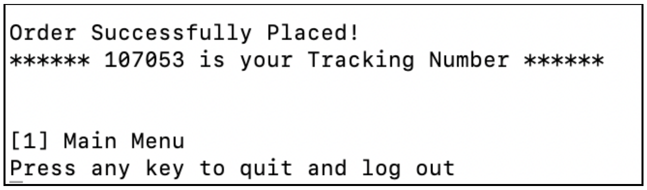
```
Hello and Welcome to the bookstore!

[1] Search for book (by Title, ISBN, Author, Genre, Rating)
[2] View Cart
[Q] to quit and log out
2
               ISBN                              Title Author FirstName Author LastName        Genre  Pages Rating  Price  Stock     Format
0  989-28-79-82749-6           9803 North Millworks Road          Carolyn           Segal      Mystery    384   2.00  22.99     50  Hardcover
0  989-28-3705-633-3             It's Never Just a Glass          Leonard         Nabokov  Young Adult    222   1.00  19.99     30  Hardcover
0  989-28-654-2017-5  Inconvenient Confessions: a memoir           Oliver           Lowry       Memoir    337   4.00  29.99     26  Paperback

[1] Checkout
[2] Continue Shopping
```

- Checkout
    - If they choose to checkout, they can choose to deliver to the address on file, or create a new shipping address



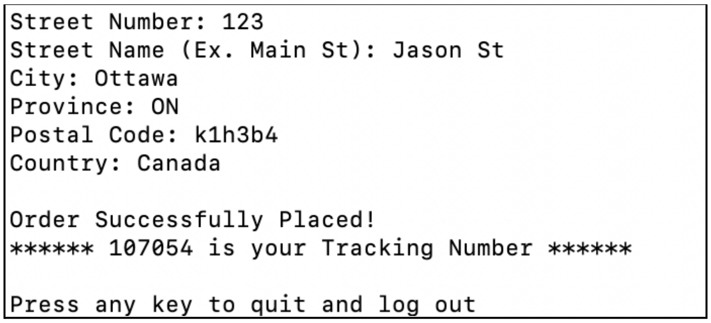
```
[1] Ship order to address on file:  123 Sesame Street Ottawa ON K2H8A7 Canada
[2] Ship to new address
```

    - If they choose to ship to an address on file, their order will be completed, and they are given a tracking number



```
Order Successfully Placed!
****** 107053 is your Tracking Number ******


[1] Main Menu
Press any key to quit and log out
```

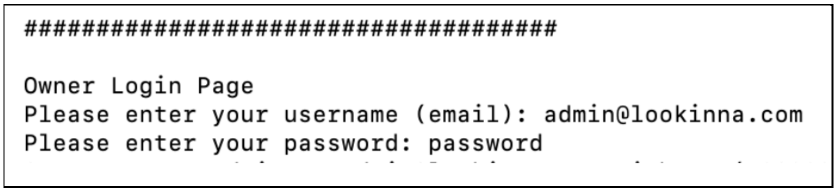
    - If they choose to add a new address, they are sent to the following screen, where they enter the new shipping details



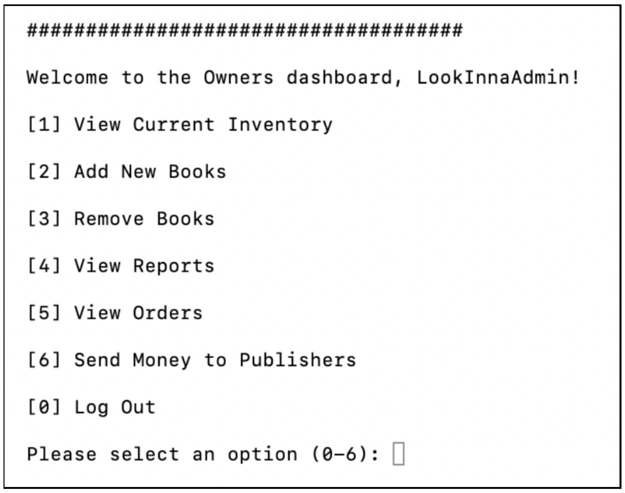
```
Street Number: 123
Street Name (Ex. Main St): Jason St
City: Ottawa
Province: ON
Postal Code: k1h3b4
Country: Canada

Order Successfully Placed!
****** 107054 is your Tracking Number ******

Press any key to quit and log out
```

Owner's Interface:

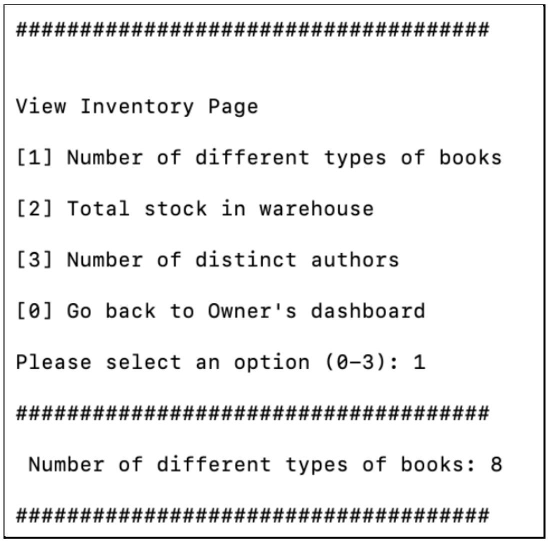
- *Owner Login Page*
    - When the owner login page is selected from the main landing page, the owner is redirected to the owner login page. Where the owner is asked for email and password. After verifying the owners email and password do they get redirected to the owners dashboard page



```
######################################

Owner Login Page
Please enter your username (email): admin@lookinna.com
Please enter your password: password
```

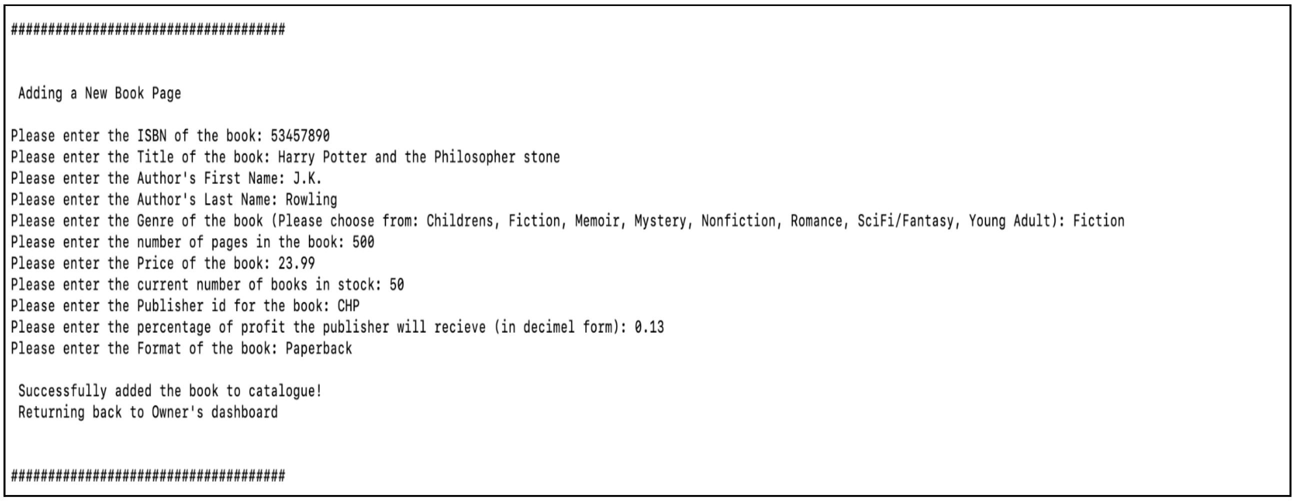
- *Owner Dashboard Page*
    - When the owner arrives at the dashboard page they are given 7 options to choose from. Depending on their selection does the owner get redirected.
    - 

```
#####################################

Welcome to the Owners dashboard, LookInnaAdmin!

[1] View Current Inventory

[2] Add New Books

[3] Remove Books

[4] View Reports

[5] View Orders

[6] Send Money to Publishers

[0] Log Out

Please select an option (0-6):
```

- *Owner - View current inventory page (with option selected)*
    - In the view inventory page, the owner can choose one of the three options. And the values will be displayed to the owner.
    - 

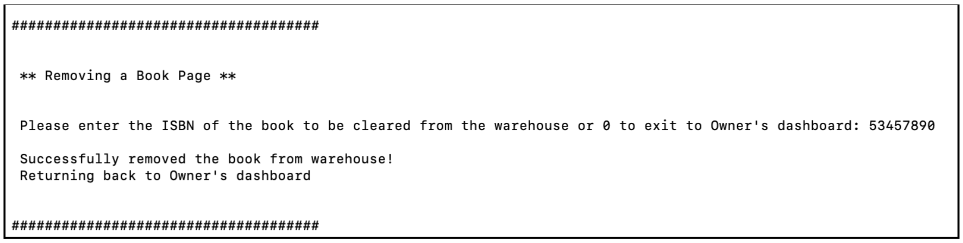
```
#####################################


View Inventory Page

[1] Number of different types of books

[2] Total stock in warehouse

[3] Number of distinct authors

[0] Go back to Owner's dashboard

Please select an option (0-3): 1

#####################################

 Number of different types of books: 8

#####################################
```

- *Owner - Add new Book page (with example shown)*
    - In the add a new book page, the owner is able to enter the details for the new book starting with the isbn of the book, title, authors name. For the genre, the owner has to enter one of the 8 genres, if not an error is thrown and the owner is told that "they failed to add a new book and to check input values". That also goes for number of pages, price, stock and percentage of profit since they all expect integer values. For publisher id, if a publisher id that is not in the publisher table is entered an error message that "This publisher is not authorised to sell. Cannot add book, returning back to Owner's dashboard!" is thrown. Otherwise the book is successfully added!



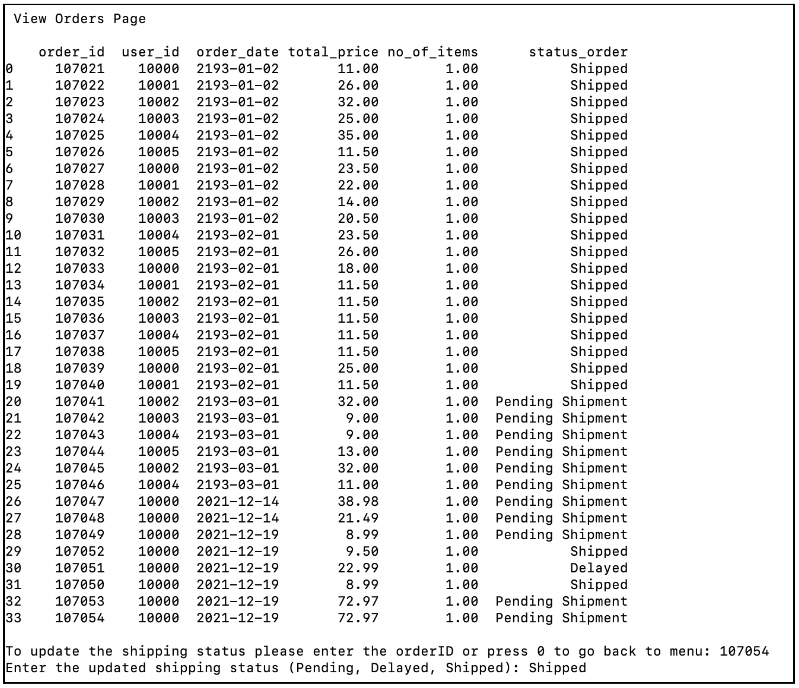
```
#####################################


 Adding a New Book Page

Please enter the ISBN of the book: 53457890
Please enter the Title of the book: Harry Potter and the Philosopher stone
Please enter the Author's First Name: J.K.
Please enter the Author's Last Name: Rowling
Please enter the Genre of the book (Please choose from: Childrens, Fiction, Memoir, Mystery, Nonfiction, Romance, SciFi/Fantasy, Young Adult): Fiction
Please enter the number of pages in the book: 500
Please enter the Price of the book: 23.99
Please enter the current number of books in stock: 50
Please enter the Publisher id for the book: CHP
Please enter the percentage of profit the publisher will recieve (in decimel form): 0.13
Please enter the Format of the book: Paperback

 Successfully added the book to catalogue!
 Returning back to Owner's dashboard


#####################################
```

- *Owner - Remove Book page (with example shown)*
    - In the remove a book page, owners are prompted to enter the isbn of the book they wish to remove. If the book's isbn is not present a message "This book does not exist in current inventory. Please try again!" is shown to the owner. Else is the correct isbn is entered the book is removed successfully with all its stock from the warehouse as well



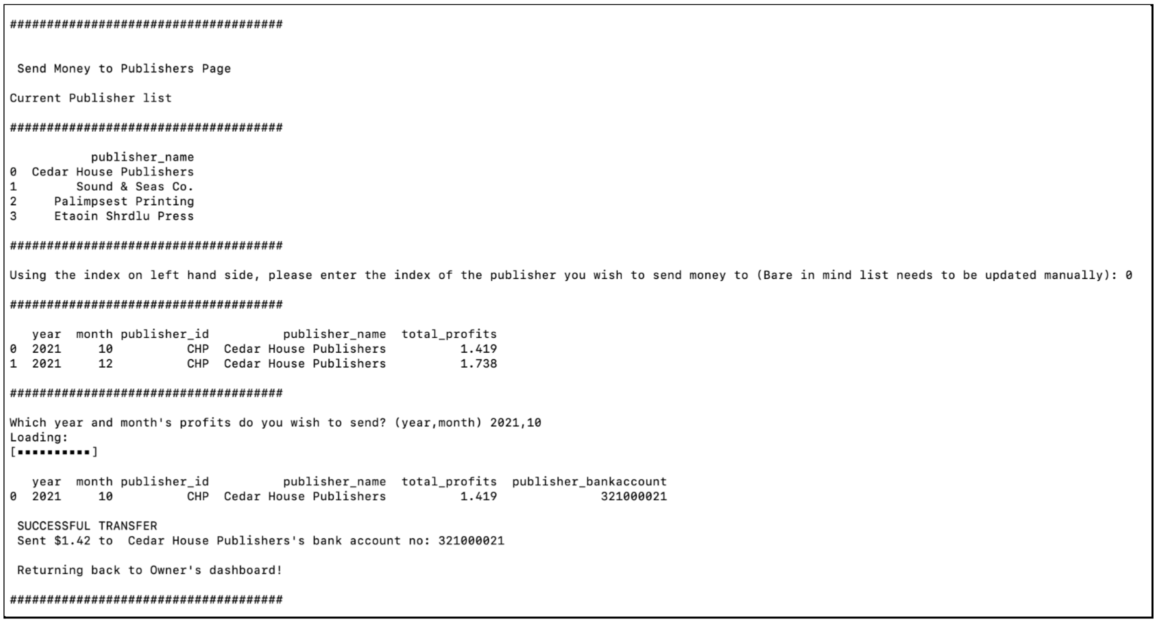
```
#####################################


 ** Removing a Book Page **


 Please enter the ISBN of the book to be cleared from the warehouse or 0 to exit to Owner's dashboard: 53457890

 Successfully removed the book from warehouse!
 Returning back to Owner's dashboard


#####################################
```

- *Owner - View Orders page (with example shown)*
    - In the view orders page, the owner is shown all orders placed to date with their date and the status of the order in shipment. Owners are then prompted if they wish to update the order status of a particular order. To update the order, the owner enters the orderID, and the status



```
View Orders Page

    order_id  user_id  order_date total_price no_of_items      status_order
0     107021    10000  2193-01-02       11.00        1.00           Shipped
1     107022    10001  2193-01-02       26.00        1.00           Shipped
2     107023    10002  2193-01-02       32.00        1.00           Shipped
3     107024    10003  2193-01-02       25.00        1.00           Shipped
4     107025    10004  2193-01-02       35.00        1.00           Shipped
5     107026    10005  2193-01-02       11.50        1.00           Shipped
6     107027    10000  2193-01-02       23.50        1.00           Shipped
7     107028    10001  2193-01-02       22.00        1.00           Shipped
8     107029    10002  2193-01-02       14.00        1.00           Shipped
9     107030    10003  2193-01-02       20.50        1.00           Shipped
10    107031    10004  2193-02-01       23.50        1.00           Shipped
11    107032    10005  2193-02-01       26.00        1.00           Shipped
12    107033    10000  2193-02-01       18.00        1.00           Shipped
13    107034    10001  2193-02-01       11.50        1.00           Shipped
14    107035    10002  2193-02-01       11.50        1.00           Shipped
15    107036    10003  2193-02-01       11.50        1.00           Shipped
16    107037    10004  2193-02-01       11.50        1.00           Shipped
17    107038    10005  2193-02-01       11.50        1.00           Shipped
18    107039    10000  2193-02-01       25.00        1.00           Shipped
19    107040    10001  2193-02-01       11.50        1.00           Shipped
20    107041    10002  2193-03-01       32.00        1.00  Pending Shipment
21    107042    10003  2193-03-01        9.00        1.00  Pending Shipment
22    107043    10004  2193-03-01        9.00        1.00  Pending Shipment
23    107044    10005  2193-03-01       13.00        1.00  Pending Shipment
24    107045    10002  2193-03-01       32.00        1.00  Pending Shipment
25    107046    10004  2193-03-01       11.00        1.00  Pending Shipment
26    107047    10000  2021-12-14       38.98        1.00  Pending Shipment
27    107048    10000  2021-12-14       21.49        1.00  Pending Shipment
28    107049    10000  2021-12-19        8.99        1.00  Pending Shipment
29    107052    10000  2021-12-19        9.50        1.00           Shipped
30    107051    10000  2021-12-19       22.99        1.00           Delayed
31    107050    10000  2021-12-19        8.99        1.00           Shipped
32    107053    10000  2021-12-19       72.97        1.00  Pending Shipment
33    107054    10000  2021-12-19       72.97        1.00  Pending Shipment

To update the shipping status please enter the orderID or press 0 to go back to menu: 107054
Enter the updated shipping status (Pending, Delayed, Shipped): Shipped
```

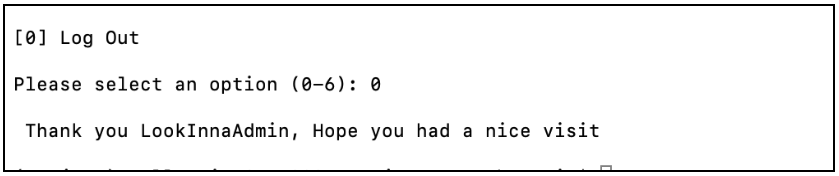
- *Owner - Send Money to Publishers page (with example shown)*
    - In the send money page, owners are prompted to select which publisher they wish to transfer their profits to. Once a correct input is entered, owners are shown the total profits publishers have received from the orders of their books for each month. Owners are then prompted to enter which year and month profits they wish to send. After entering the input they are shown a loading bar (simulating an actual bank transfer) and if correct inputs are entered they are shown a successful confirmation message and redirected to Owner's dashboard page.



```
#####################################


 Send Money to Publishers Page

Current Publisher list

#####################################

           publisher_name
0  Cedar House Publishers
1        Sound & Seas Co.
2     Palimpsest Printing
3     Etaoin Shrdlu Press

#####################################

Using the index on left hand side, please enter the index of the publisher you wish to send money to (Bare in mind list needs to be updated manually): 0

#####################################

   year  month publisher_id          publisher_name  total_profits
0  2021     10          CHP  Cedar House Publishers          1.419
1  2021     12          CHP  Cedar House Publishers          1.738

#####################################

Which year and month's profits do you wish to send? (year,month) 2021,10
Loading:
[▪▪▪▪▪▪▪▪▪▪]

   year  month publisher_id          publisher_name  total_profits  publisher_bankaccount
0  2021     10          CHP  Cedar House Publishers          1.419              321000021

 SUCCESSFUL TRANSFER
 Sent $1.42 to  Cedar House Publishers's bank account no: 321000021

 Returning back to Owner's dashboard!

#####################################
```

-

- *Owner - Log out (exits the program)*
    - If the owner decides to logout they exit the program completely with a message.



```
[0] Log Out

Please select an option (0-6): 0

 Thank you LookInnaAdmin, Hope you had a nice visit
```

2.6. Bonus Features

- Search by specific rating intervals and search by minimum rating
- Login screen separate for user and owner
- Owner can update order shipping status
- Search Book, author, with fuzzy search (approximation)
- Owner can view inventory

2.7. GitHub Repository

https://github.com/Samar20/Comp3005_Project

Citations

- **Bookstore data from:** **The Bookshop data set - Tableau**
- **Loading bar code from:** **Python how to make simple animated loading while process is running - Stack Overflow**

2.8. Appendix

* Team members are in different time zones (One in Ottawa and the other in UAE)

1. Time Slot #1: 9am
2. Time Slot #2: 10am
3. Time Slot #3: 11am