DataBase

Project Report

Bookstore System

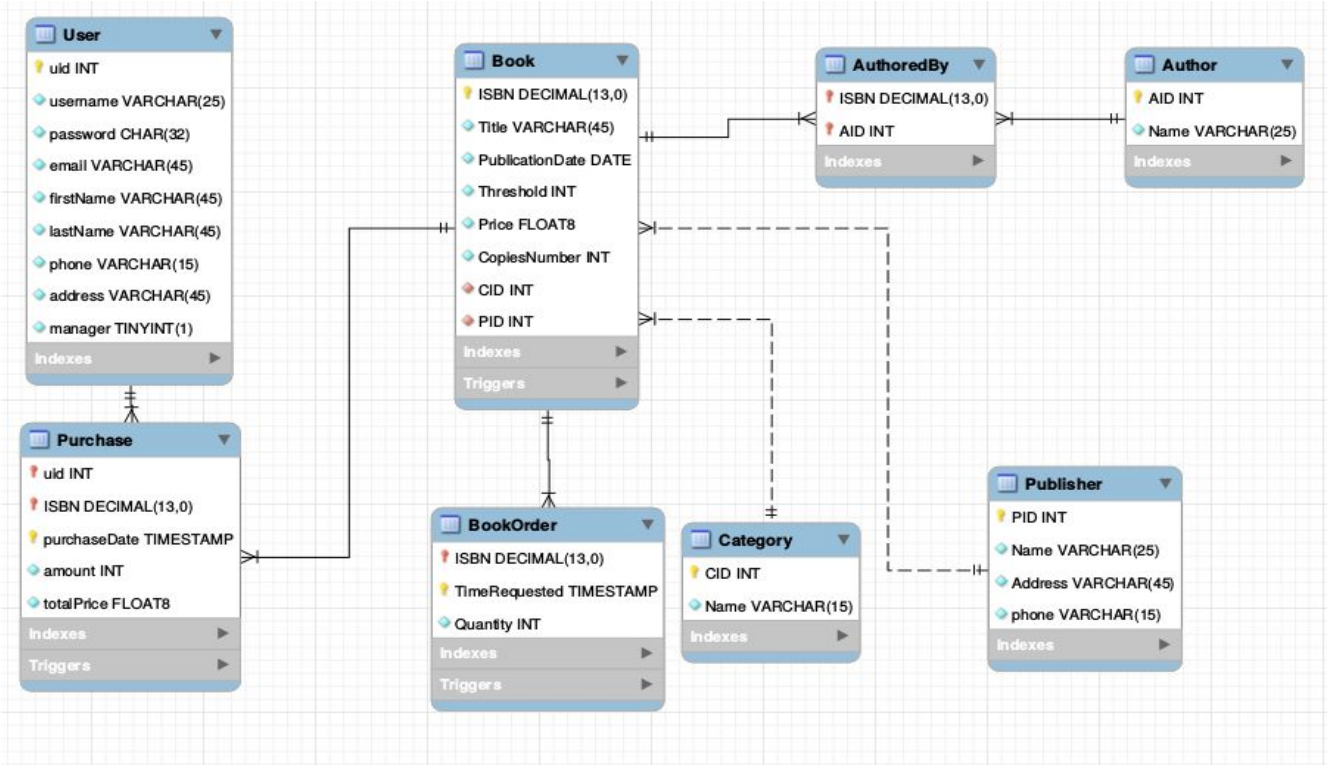
Team

Amr Mohamed NASR ELDIN	45
Michael Raafat MIKHAIL	54
Marwan MORSY	69

I. Objective :

It is required to analyze, design and implement a database system to support the operations of a simplified online bookstore. Also, It is required to create the necessary forms to perform these operations.

II. The database ERD diagram :



III. Detailed Analysis Of Database :

- ## Overall Design

Categories were made into a seperate table since many might have the same category so to make updates consistent, we only add a foreign key to the book to a category. Publisher was made into its own entity too like categories. Authors might write multiple books and the same book can have multiple authors so logically we separated them to keep updates consistent.

Users and books are related by entity of purchases, which indicates the different purchase of users from book. This entity allows us to have history of sales and purchase for both users and book, as well as have statistics of how

many books were sold and how much was gained. The last entity of orders indicates the different orders of each books and the quantity requests. Attributes like no of copies of books and author id and other were made unsigned when suitable.

- ### Triggers

A trigger Before Delete on Orders : Assumed to be a confirmed order which updates the book and adds the amount of the order to the suitable book number of copies.

A trigger Before Update on Books : If the new value for number of copies is going to be less than 0 then stop transaction.

A trigger After Update on Books : If number of copies falls under threshold, add an order that compensates the difference between them.

A trigger Before Insert on Purchase : When a purchase is inserted, we must reduce the book copies of the book purchased by the purchase amount. If not possible stop operation.

- ### Indexes

Index on category name since it is unique to help in search and updates.

Index on publisher name since it is unique to help in search and updates.

Index on author name since it is unique to help in search and update.

Index on cid,pid, and title in the book table to help us when searching on books.

Index on author id in the authoredBy table to help us when searching for specific authors.

Indexes on username and email in User table to help us in log in, registration and search operations.

Indexes on isbn and uid in Purchase table to help us in getting statistiques needed for report generation.

- ### Transactions

Most of the simple operations as inserting, deleting and updating that were simple were done in its own transaction like in tables like publisher, author, category, order and user.

A Transaction was made in purchasing a cart, as only when we can successfully add all purchases required of cart we commit, otherwise we rollback. This is found in method handleCart in class PurchasesDAO.

A Transaction was made when adding a new book where we only commit if the book itself is added to book table, and all additions to authored by to

indicate its authors are successful. Otherwise, rollback. Found in addBook in BookDataDAO.

A Transaction was made when updating a book where only when the book itself is modified, all of its changed authors are actually changed in authoredBy do we commit, otherwise rollback. Found in updateBook in BookDataDAO.

IV. Code Modules (WEB):

- ## Bean package

Package containing the basic models for the entities of the database.

- ## Config Package

Package containing the configuration for security filter, which url are allowed to be served only to users, and which ones only to manages.

- ## Dao Package

Package containing the code linking the database commands with the code, it acts as the access point to the database by providing functions that include all the sql commands needed and executing it, it protects against code injections and will return the suitable beans for each request. It contains separate modules for each type of entity operations, and includes transaction logic in it.

- ## Filter Package

Package containing the filter responsible for intercepting all requests and verifying that the request has the permission to access and get served from the server using the security configuration, and either allowing the request through or blocking it and returning a suitable response.

- ## Provider Package

Package containing the provider interface which has the database information needed for connection and a provider class that provides a connection that will be used in all DOA classes.

- ## Request Package

Package containing a wrapper class for the request that contains the user role of the request.

- ## Servlet Package

Package containing the different servlets that serve their suitable requests.

AuthorManagerServlet.java : serves all manager operations on authors table (add, edit, search).
LogoutServlet.java : serves logout requests.
AutoCompleteServlet.java : serves autocomplete requests for any data.
ManagerServlet.java : serves the manager page.
BookManagerServlet.java : serves all manager operations on books table(search, add, edit).

OrderManagerServlet.java : serves all manager operations on orders table(search, add, confirm).
BookServlet.java : serves user search requests on books table.
PublisherManagerServlet.java : serves all manager operations on publisher table(search, add, edit).
CartServlet.java : serves all user requests regarding additions, removals and showing carts.
PurchaseServlet.java : serves user checkout request of the cart.
CategoryManagerServlet.java : serves all manager operations on category table(search, add, edit).
RegistrationServlet.java : serves actual user registration requests if post request, and serves the actual registration page if get.
EditUserServlet.java : serves all user requests regarding changes in his profile.
ReportServlet.java : serves all report requests by returning the report as an encoded base 64 string of the suitable report pdf.
HomeServlet.java: serves the home page.
UserManagerServlet.java : serves all manager operations on user table(search, promote).
LogInServlet.java : serves the login requests if post, serves login page if get.
UserServlet.java : serves the user page.

- ## Utils Package

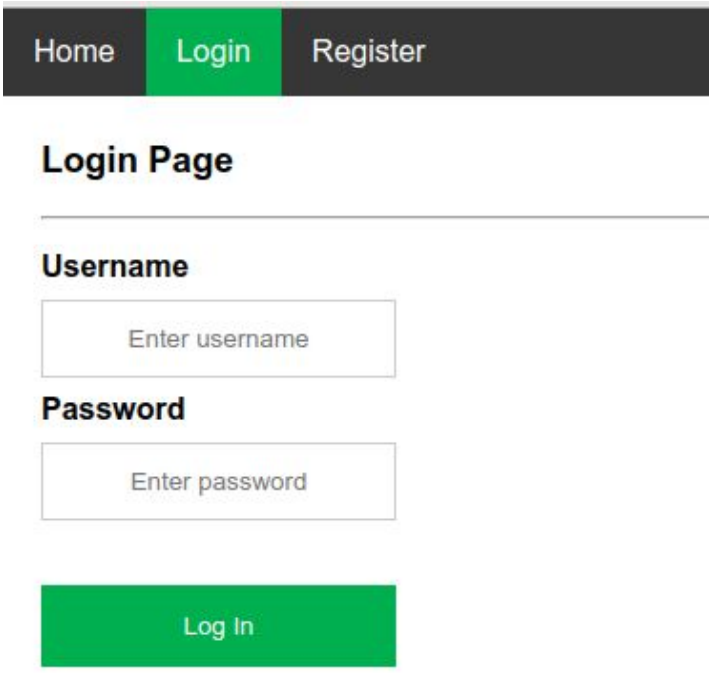
AppUtils.java : responsible of storing and retrieving session variables, and redirection urls.
PaymentUtils.java : responsible of simulating a payment gateway.
UrlPatternUtils.java : responsible of parsing and verifying url strings.
ParseUtils.java : responsible of some minor verification operations on parameters passed from the user requests.
SecurityUtils.java : responsible of handling some security checks to see if a request needs to have permission for the page it wants or if it even needs one.

- ## Non-Source-File Used

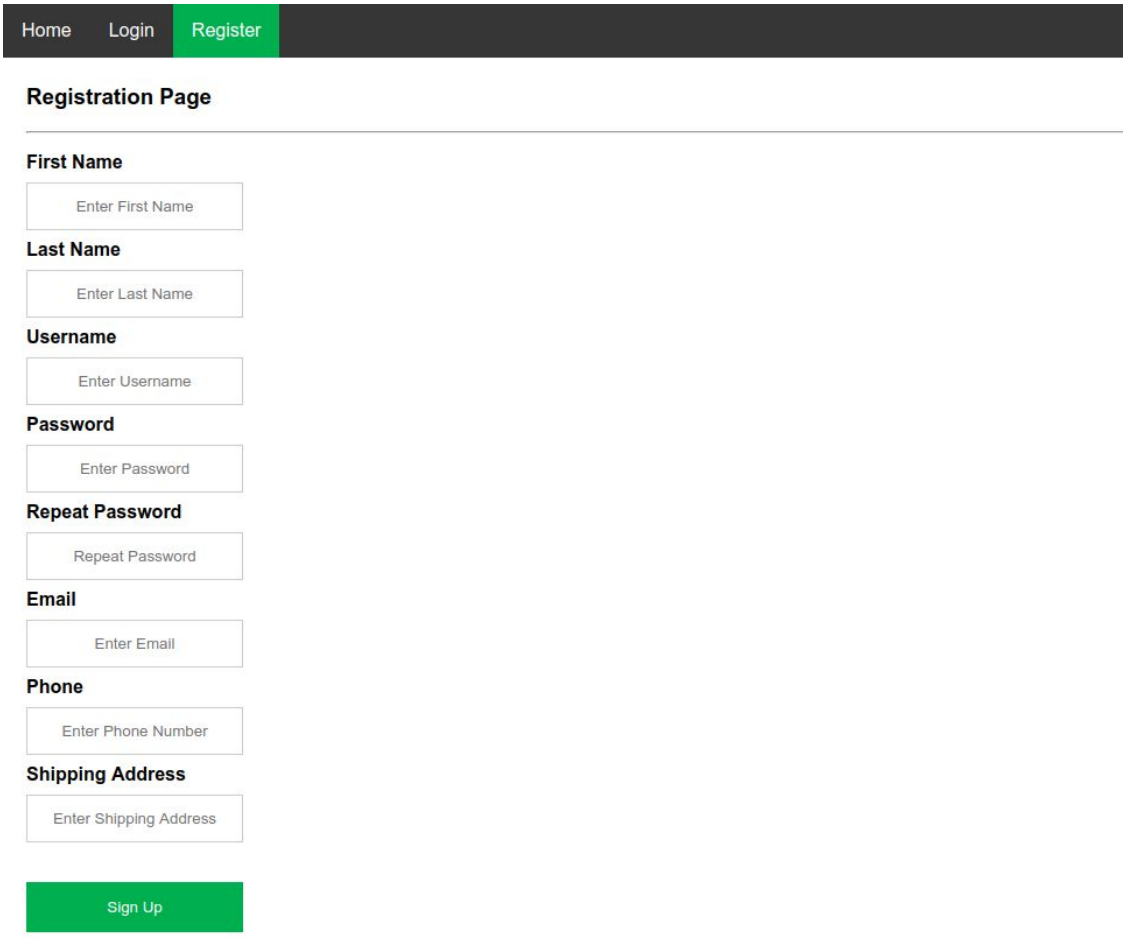
Datapopulator.jar : populates our database with 6 categories, a root user with password as ‘root’ and username ‘root’, potentially 500,000 author and 500,000. Then inserts 1,000,000 book with each book having up to 5 different authors. This jar was made by us but its source code is omitted as it is not seen as part of the project.
Reports_generation.sql : contains sql commands used in jasper reports.
Create_bookstore.sql : the database creation script containing all table creations, indexes definitions and triggers definitions.

V. Sample Screen shots :

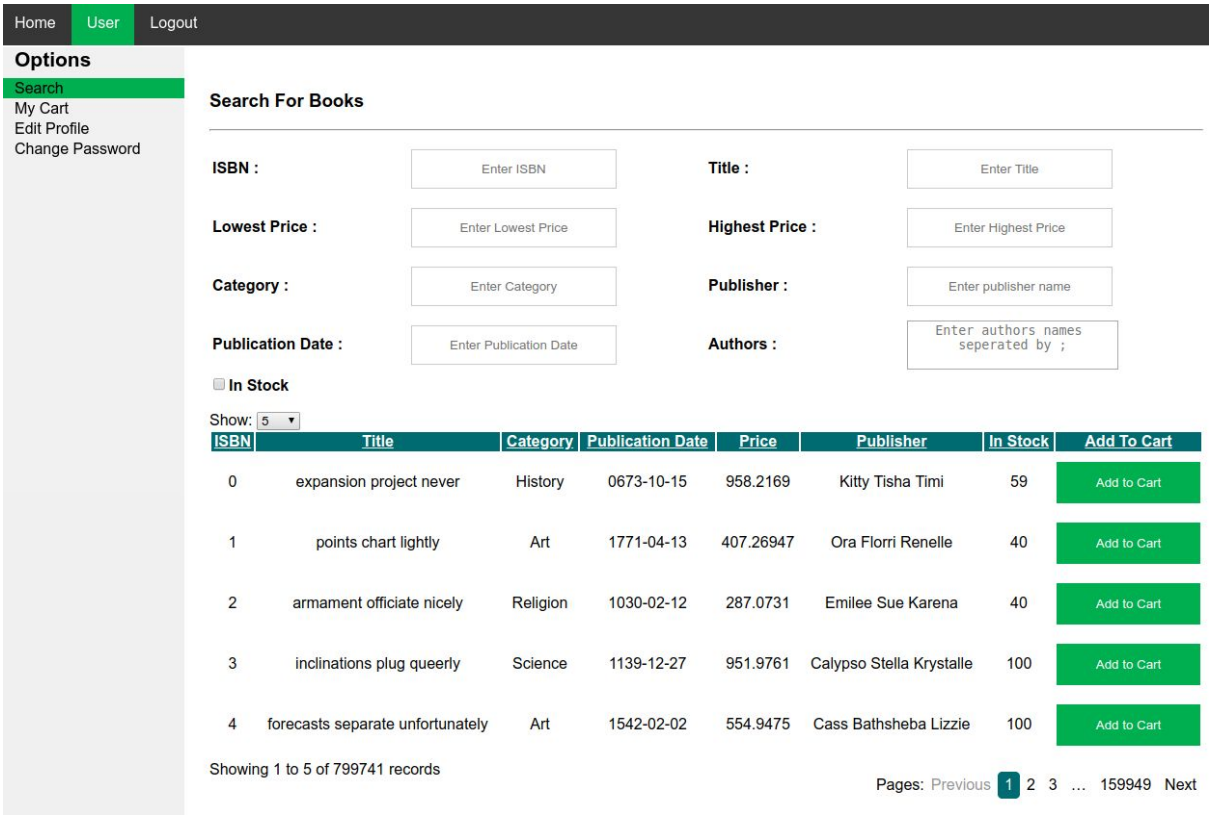
- **Sign in**



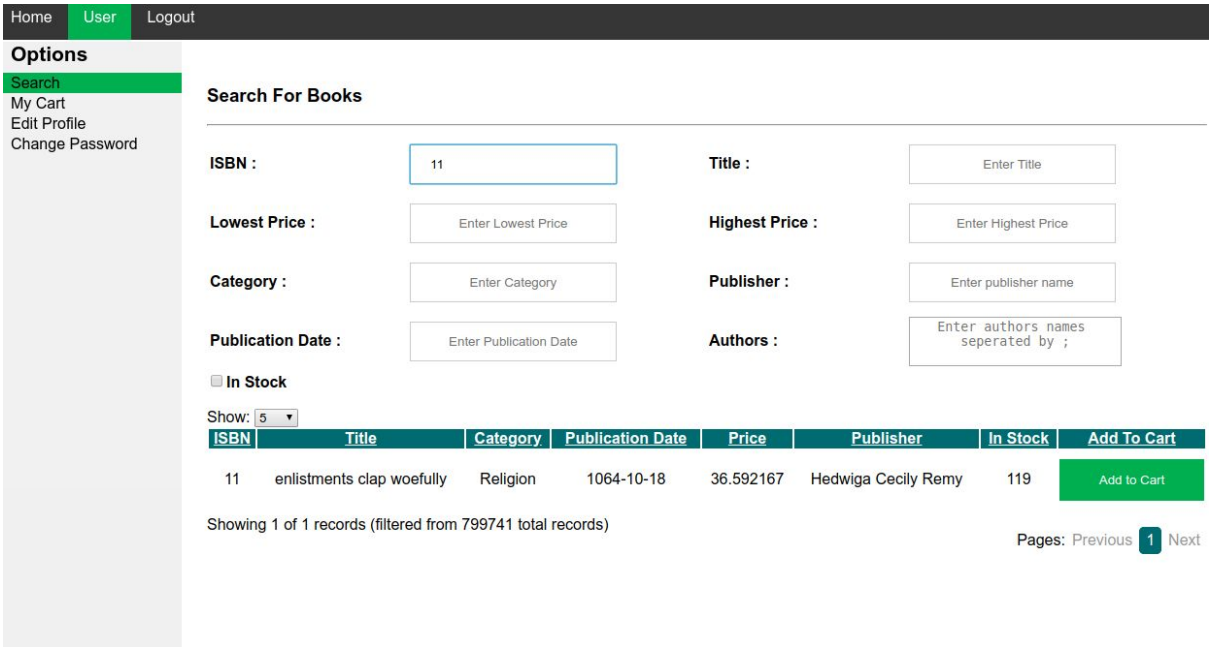
- **Sign up**



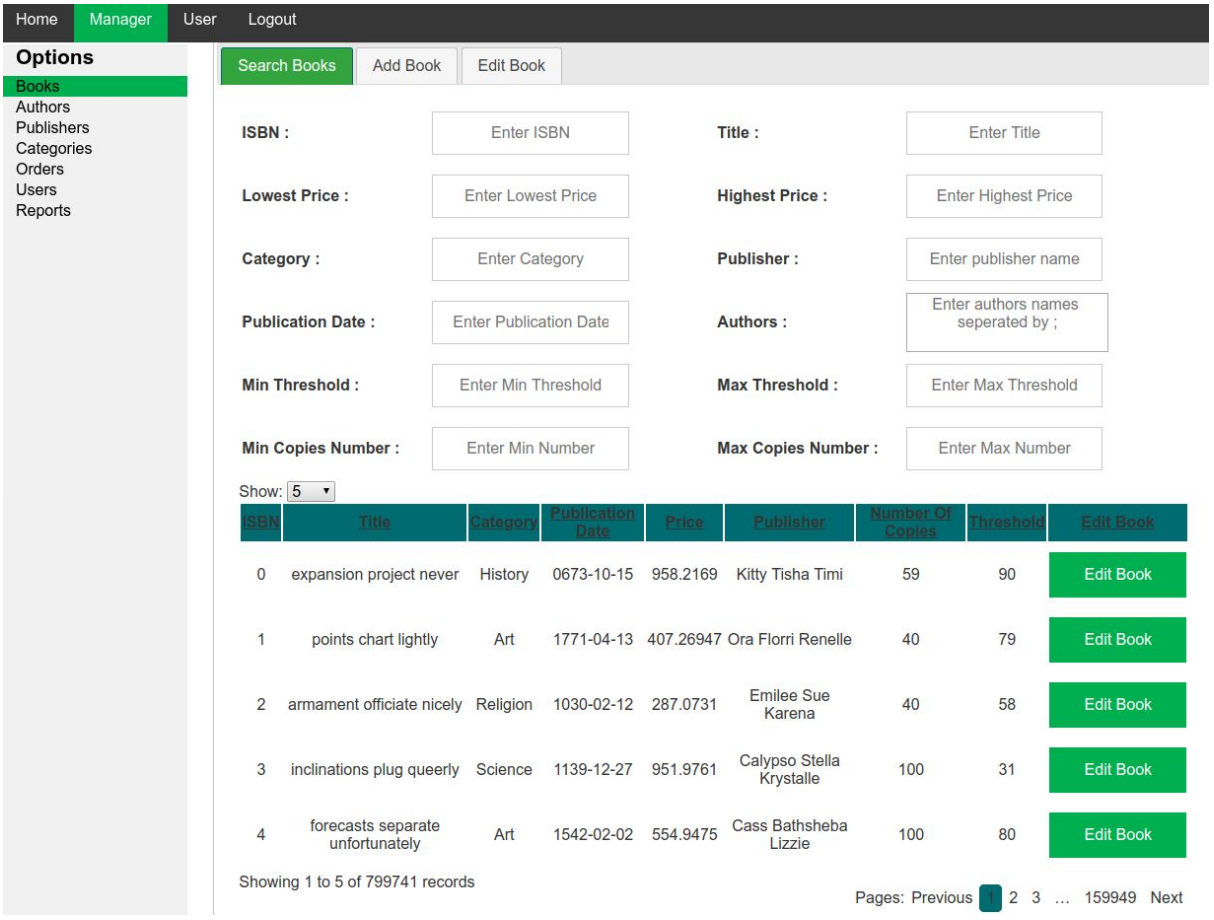
- ## Search view user side



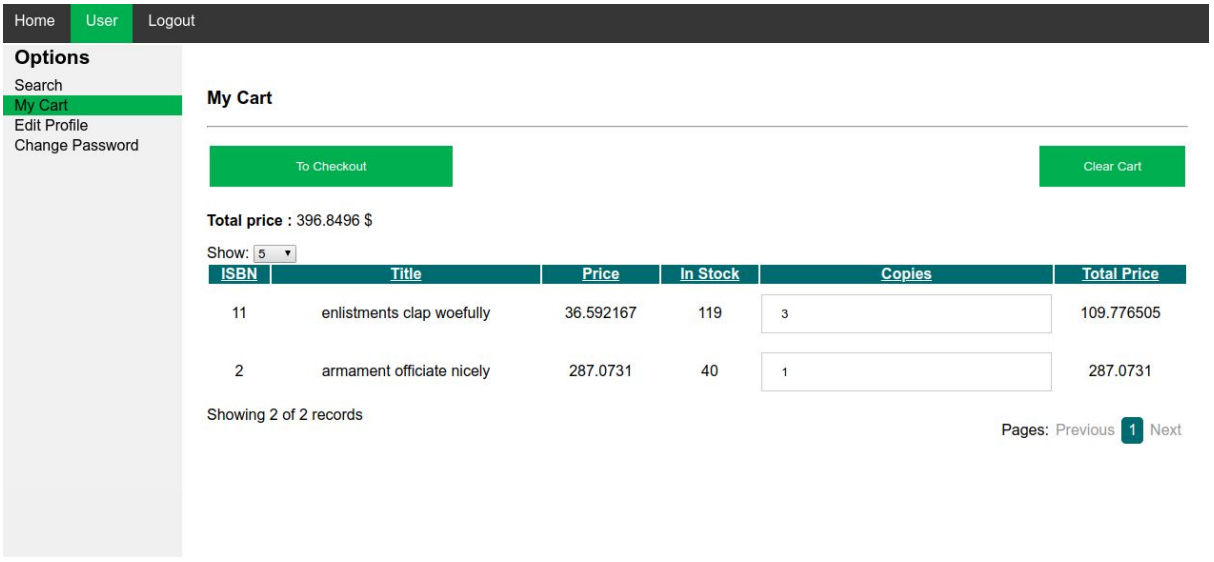
- ## Full search view user side



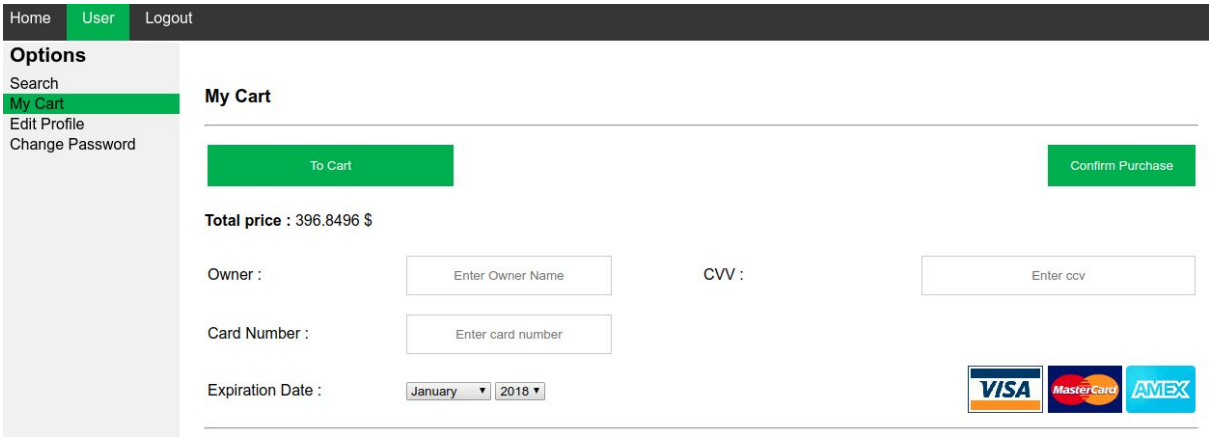
- ## Search view for Manager side



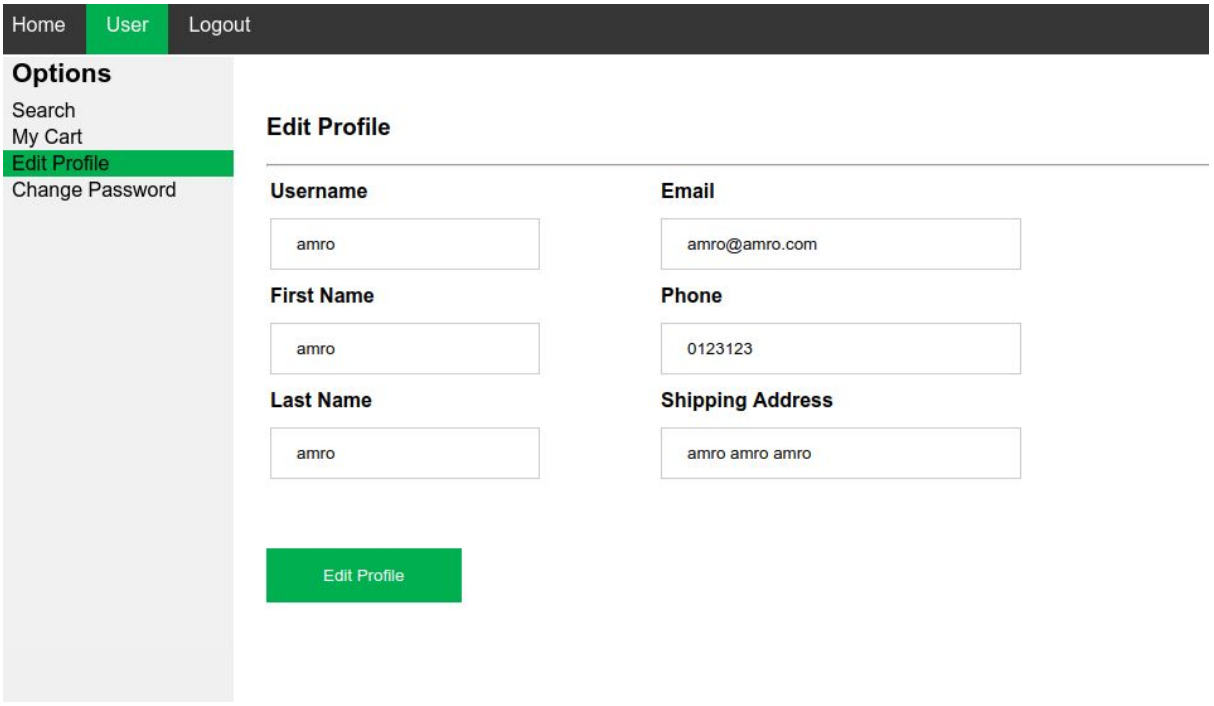
- ## My cart view

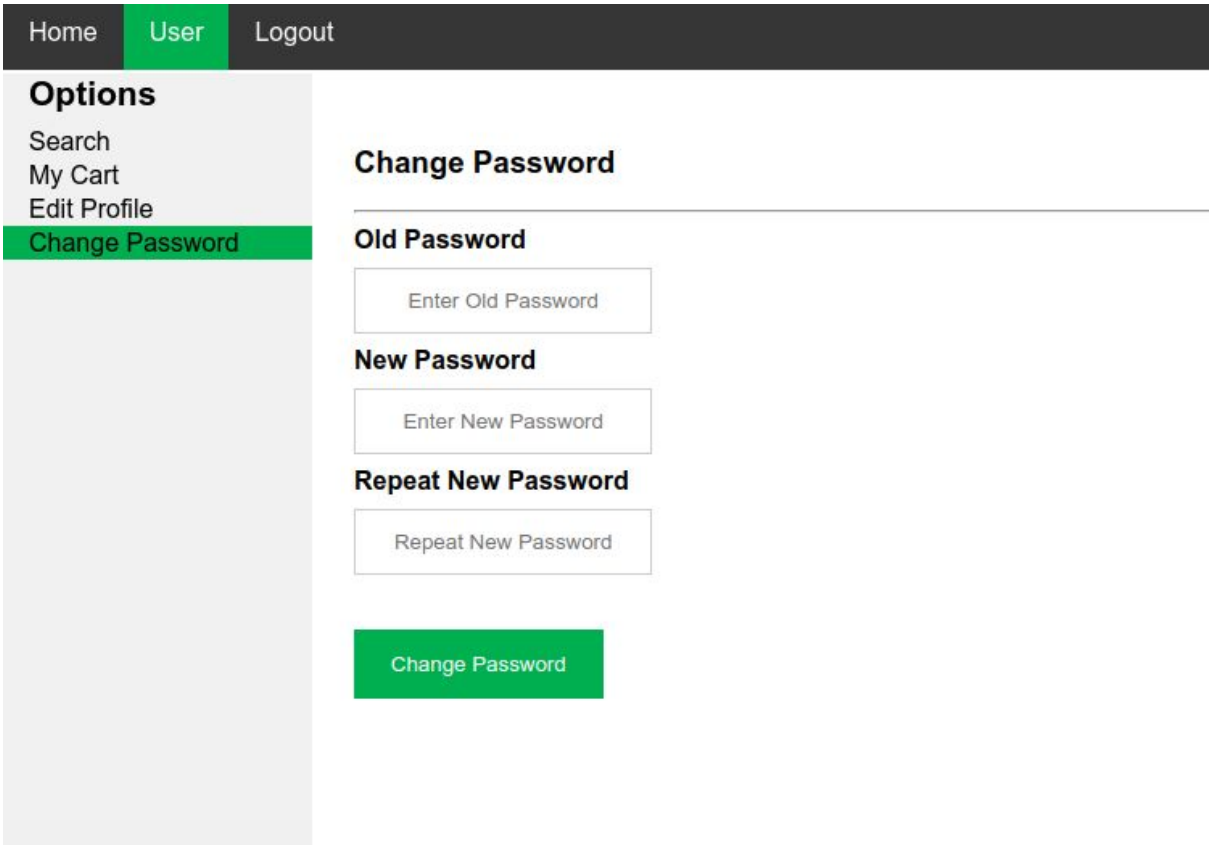


- **Checkout view :**

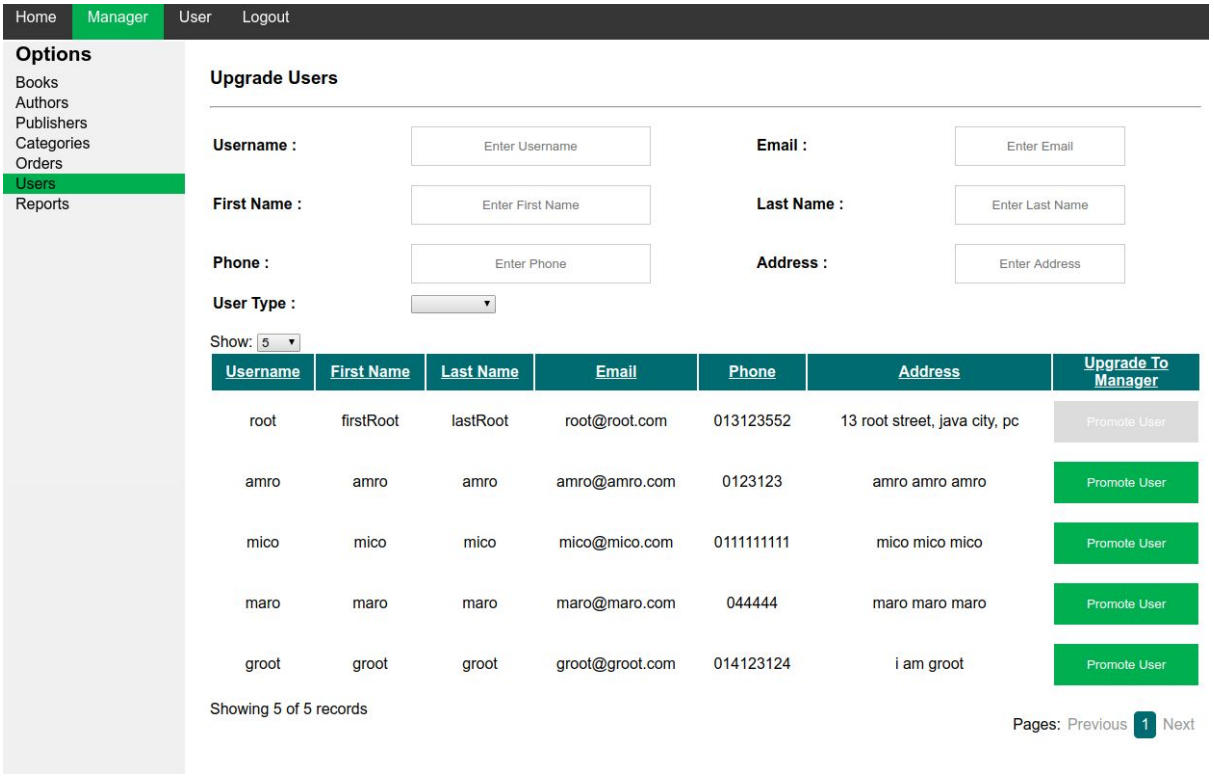


- **Edit profile view :**

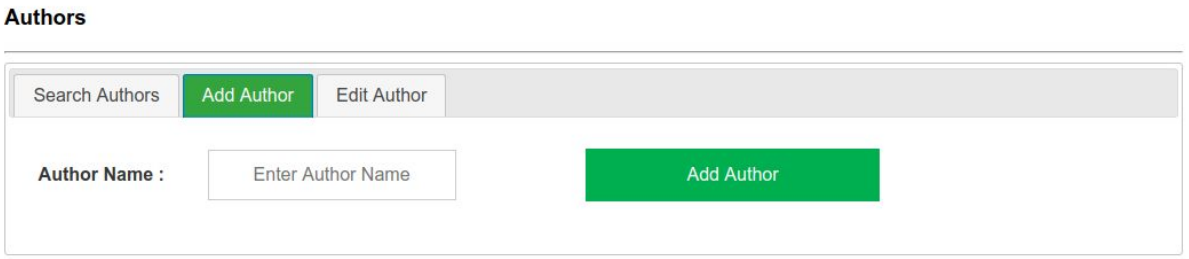




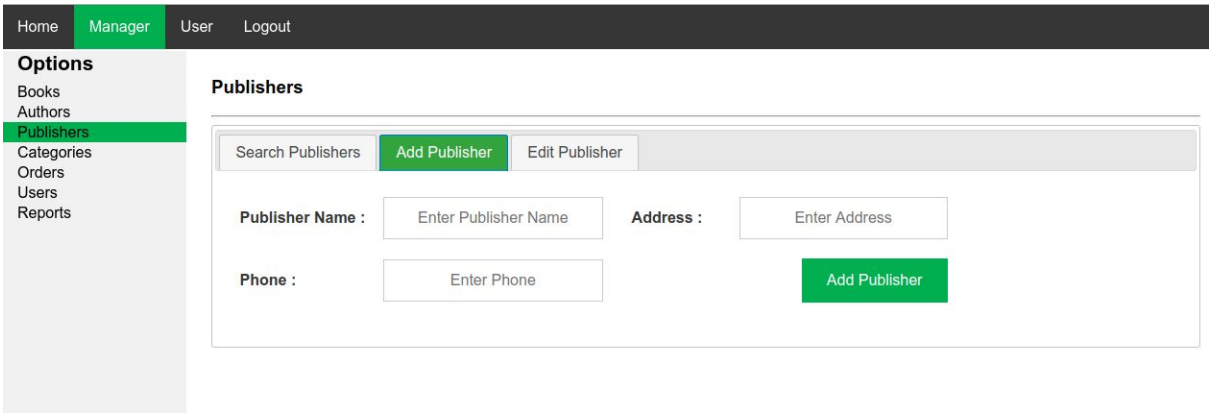
- **Manager promote user view**



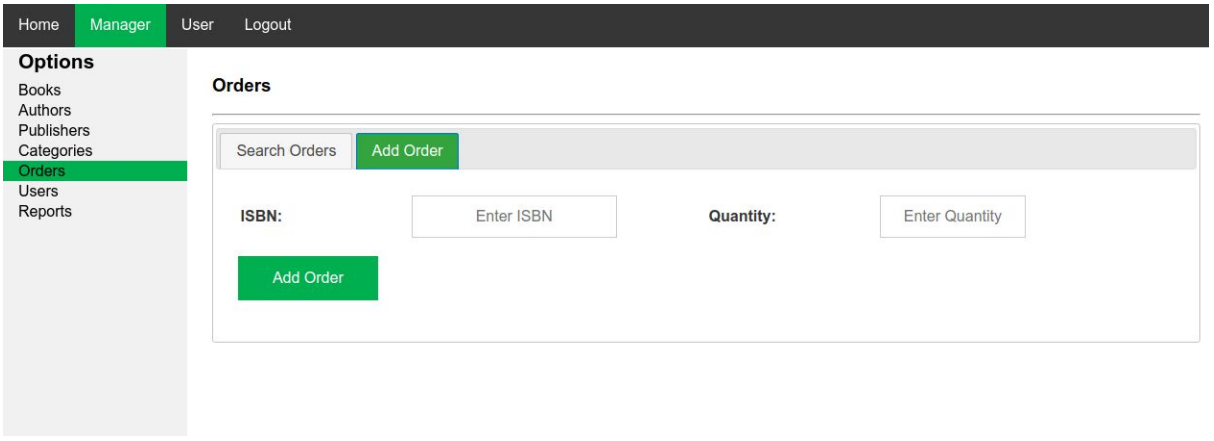
- **Manager add author view**



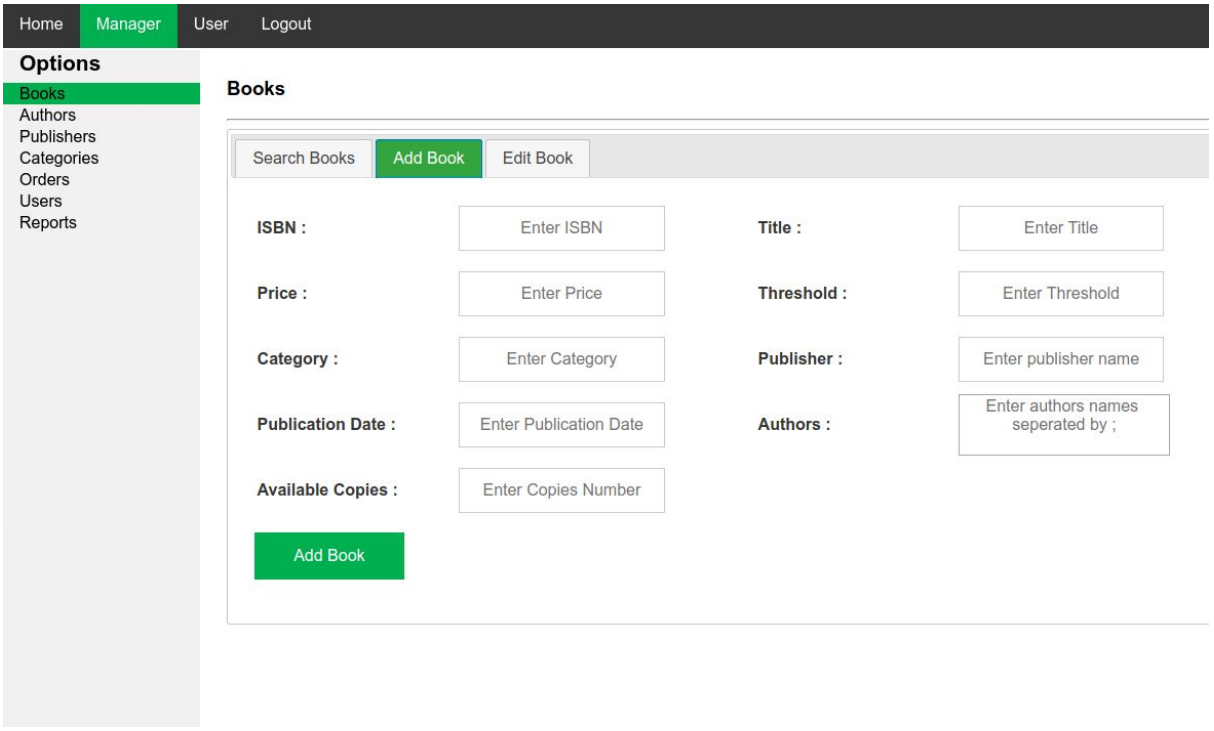
- **Manager add publisher view**



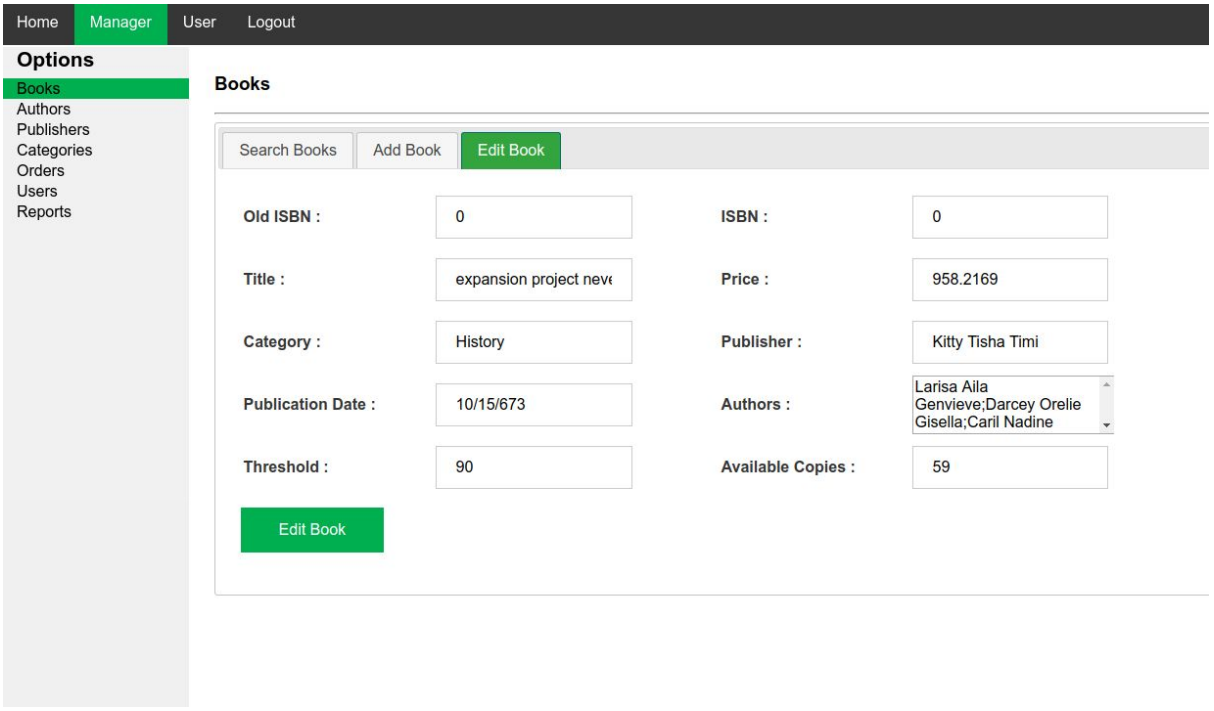
- **Manager add order view :**



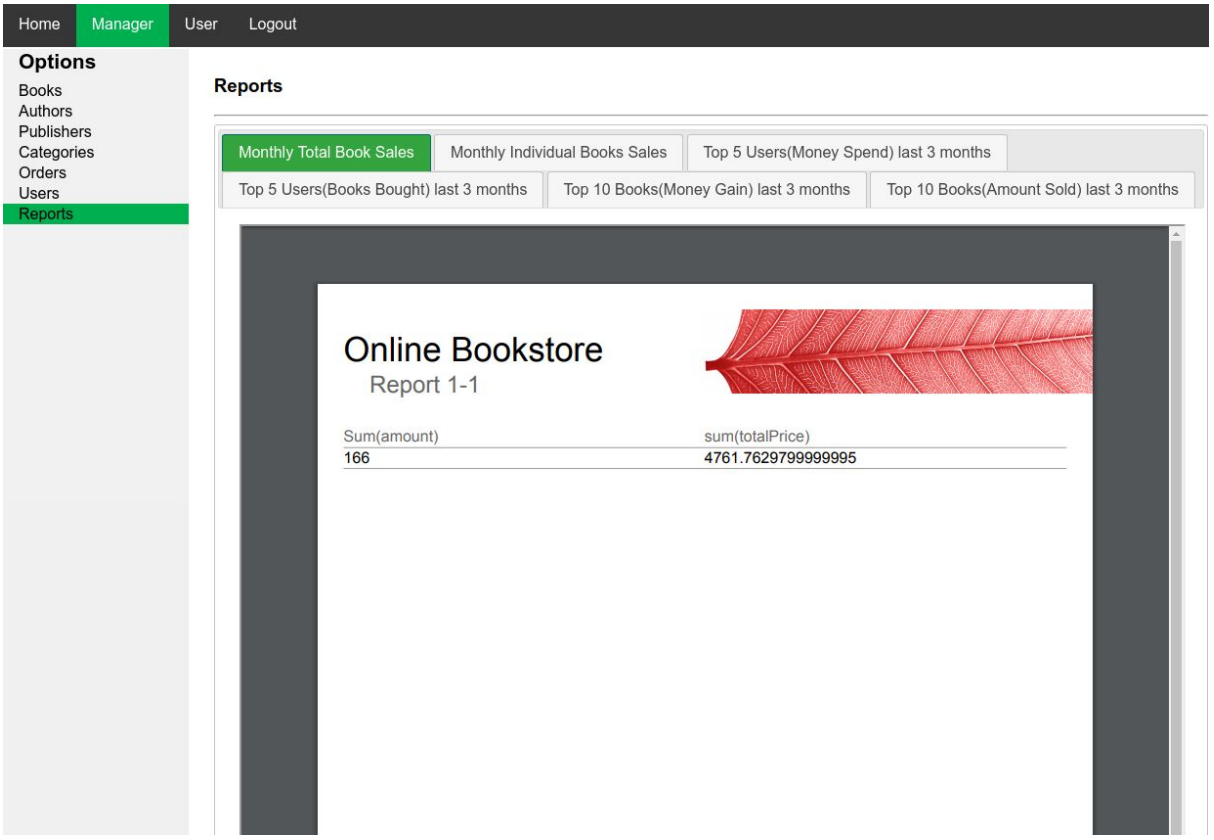
- **Manager add book view :**



- **Manager edit book view :**



- **Report View**


Home
Manager
User
Logout
Options
Books
Authors
Publishers
Categories
Orders
Users
Reports
Reports
Monthly Total Book Sales
Monthly Individual Books Sales
Top 5 Users(Money Spend) last 3 months
Top 5 Users(Books Bought) last 3 months
Top 10 Books(Money Gain) last 3 months
Top 10 Books(Amount Sold) last 3 months
Online Bookstore
Report 1-1
Sum(amount)
sum(totalPrice)
166
4761.7629799999995