

**IDB-BISEW IT Scholarship Project**

**A Project Documentation**

**on**

**Library Management System**

**SUBMITTED TO**

**Sayed Zahidul Hassan**

Consultant (.NET),

Show & Tell Consulting Ltd.

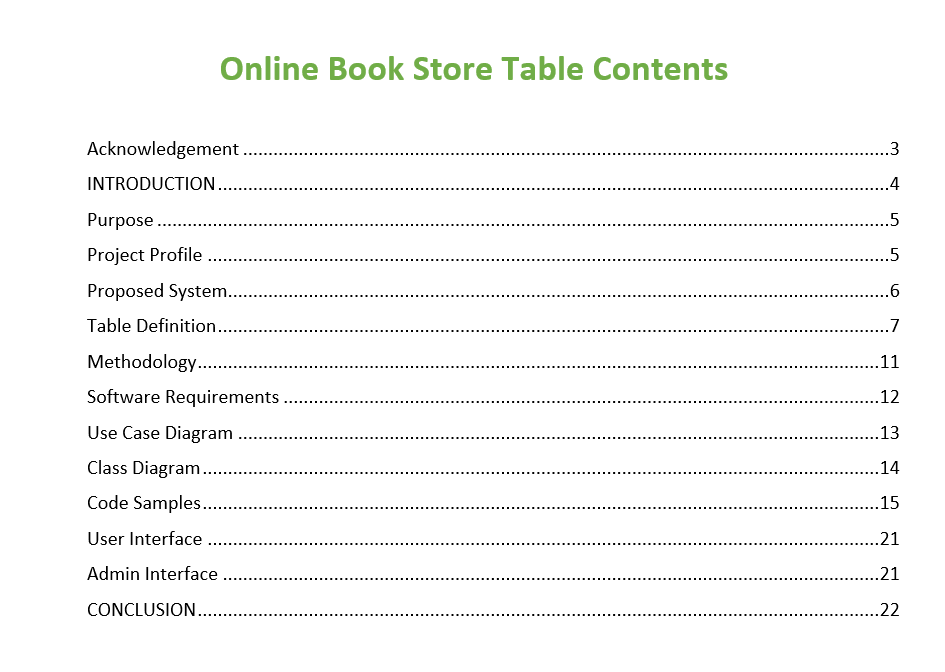
**Supervised By**

**Md. Foysal Wahid**

Trainer, New Vision Information Technology Ltd. (NVIT)

**SUBMITTED BY**

|  |  |
| --- | --- |
| ID | NAME |
| 1272311 | Sayed Minhajur Rahman (Leader) |
| 1272015 | Jahidul Islam |
| 1271633 | Iftekhar Ahmed |
| 1272262 | Tanvir Ehsan Bhuiyan |
| 1271556 | Mohammed Yousof |
| 1271801 | Md. Gias Uddin |



INTRODUCTION



Now a day’s people are too much busy, they just want everything by one click sitting in the room. Through this perspective we just prepare this system so that user can purchase a book through online instead of going out to a bookstore. The system aims to reduce time associated with conducting purchase any book with anyone requirements.

This project ‘Online Bookstore’ has been developed on MVC Core API, Angular 6, Web API, SQL Server 2016. The vision of this project is to sell book through online.

An online book store is a virtual store on the Internet where customers can search, select and buy books of interest. The selected books may be collected in a shopping cart. At checkout time, the items in the shopping cart will be presented as an order. At that time, more information will be needed to complete the transaction. Usually, the customer will be asked to fill or select a billing address, a shipping address, a shipping option, and payment method may be bKash, Rocket or cash on delivery

The system includes both a front-end portal for customers and a back-end management system used by administrators to manage the website in the background.

Purpose

The main objective of the project is to create an online book store that allows users to search and purchase a book online based on title, author and publication. The selected books are displayed in a tabular format and the user can order their books online through shopping cart.

A customer can, create, sign in to his account, place items into a shopping cart and purchase using his credit card details.

Online bookstore broke the management mode of traditional bookstore, as long as you have a computer, you can buy the book anywhere, saving time and effort, shortening the time of book selection link effectively. The online bookstore system based on the principle of provides convenience and service to people.

Project Profile

**Project definition :** Online Bookstore Management

**Front-End :** Angular-6, Material-Theme, NgxBootstrap

**Back-end :** SQL Server 2016, MVC Core API, Web API

**Type of application :** Single Page Application

**Other tools :** Visual Studio 2017, MS Office 2016, Visual Studio Code

**Time duration :** 30 days.

Proposed System

* **User Perspective:**
* Nowadays, people are too busy, and they prefer convenience, so we've designed this system to allow users to purchase books online from the comfort of their own homes. The system's primary goal is to save time and simplify the process of buying books to meet individual requirements.
* For users, the proposed system is divided into two main categories:
* Search for books by various criteria such as author, category, publication, genre, and more.
* Access detailed information about authors, books, and publications.
* Registration is required for purchasing books.
* Utilize a shopping cart system that allows users to adjust the quantity of books.
* Create a Wishlist and manage its contents.
* Read books online and download PDF copies.
* Establish direct communication with the management through email or other contact methods.
* Provide guidance to users on how to navigate and use the system.
* Offer multiple payment methods for book purchases, including cash on delivery, bKash, and others.
* Manage user database, allowing users to update personal information and track their orders.

**Management Perspective:**

* From a management perspective, the system offers various features to streamline operations:
* Admins can monitor user activity.
* Manage employee information and their activities.
* Admins have the ability to create and assign roles to users.
* Search for specific roles by name and view associated actions.
* Delete unnecessary roles.
* Manage book information, including adding, updating, and deleting book details and categories.
* Respond to customer feedback.
* Maintain a purchase list and search for specific purchase information by date range.
* Check purchase, order, and payment statuses.
* Confirm placed orders.
* Generate purchase, stock, and order reports, with an option to print the reports.

Table Definition

|  |  |  |  |
| --- | --- | --- | --- |
| Sl No: | Data Name | Data type | Constraint |
| 1 | CategoryId | INT | PK, IDENTITY |
| 2 | CategoryName | NVARCHAR (50) | NOT NULL |
| 3 | DDCCode | BIT | NULL |
| 4 | IsActive | BIT | NULL |

**Entity: Category**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl No: | Data Name | Data type | Constraint |
| 1 | IsActive | BIT | NOT NULL |
| 2 | ProfileImage | NVARCHAR (MAX) | NULL |
| 4 | IsSubscribed | BIT | NOT NULL |

**Entity: ApplicationUser**

**Entity: Publisher**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl No: | Data Name | Data type | Constraint |
| 1 | PublisherId | INT | PK, IDENTITY |
| 2 | PublisherName | NVARCHAR (MAX) | NULL |
| 3 | Address | NVARCHAR (MAX) | NULL |
| 4 | Email | NVARCHAR (MAX) | NULL |
| 5 | PhoneNumber | NVARCHAR (MAX) | NULL |
| 6 | IsActive | BIT | NOT NULL |

**Entity: Author**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl No: | Data Name | Data type | Constraint |
| 1 | AuthorId | INT | PK, IDENTITY |
| 2 | FirstName | NVARCHAR (MAX) | NULL |
| 3 | LastName | NVARCHAR (MAX) | NULL |
| 4 | BirthDate | DATETIME | NULL |
| 5 | Biography | NVARCHAR (MAX) | NULL |
| 6 | Email | NVARCHAR (MAX) | NULL |
| 7 | Phone | NVARCHAR (MAX) | NULL |
| 8 | IsActive | BIT | NOT NULL |

**Entity: BookAuthor**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl No: | Data Name | Data type | Constraint |
| 1 | BookId | INT | FK (BOOK) |
| 2 | AuthorId | INT | FK (Author) |

**Entity: Book**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl No: | Data Name | Data type | Constraint |
| 1 | BookId | INT | PK, IDENTITY |
| 2 | Title | NVARCHAR (MAX) | NULL |
| 3 | ISBN | NVARCHAR (MAX) | NULL |
| 4 | PublisherId | INT | FK(Publisher) |
| 5 | PublishedYear | DATETIME | NOT NULL |
| 6 | Edition | NVARCHAR(MAX) | NULL |
| 7 | TotalCopies | INT | NULL |
| 8 | Language | NVARCHAR(MAX) | NULL |
| 9 | Description | NVARCHAR(MAX) | NULL |
| 10 | BookPrice | DECIMAL(18, 2) | NULL |
| 11 | RefundPrice | DECIMAL(18, 2) | NULL |
| 12 | DDCCode | NVARCHAR(MAX) | NOT NULL |
| 13 | IsActive | BIT | NOT NULL |
| 14 | IsDigital | BIT | NOT NULL |
| 15 | PublisherAgreement | BIT | NOT NULL |
| 16 | CategoryId | INT | FK(Category) |
| 17 | CoverFileName | NVARCHAR(MAX) | NULL |
| 18 | EBookFileName | NVARCHAR(MAX) | NULL |
|  | AgreementFileName | NVARCHAR(MAX) | NULL |

**Entity: BookCopy**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl No: | Data Name | Data type | Constraint |
| 1 | BookCopyId | INT | PK, IDENTITY |
| 2 | CallNumber | NVARCHAR(MAX) | NULL |
| 3 | IsAvailable | BIT | NOT NULL |
| 4 | IsActive | BIT | NOT NULL |
| 5 | condition | ENUM(BookCondition) | NUT) |
| 6 | BookId | INT | FK(BOOK) |
|  | DDC | NVARCHAR(MAX) | NULL |
|  | ShelfId | INT | FK(BOOK) |

**Entity: BookFloor**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl No: | Data Name | Data type | Constraint |
| 1 | BookFloorId | INT | PK, IDENTITY |
| 2 | BookFloorName | NVARCHAR(MAX) | NULL |
| 12 | IsActive | BIT | NOT NULL |

**Entity: BookRack**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl No: | Data Name | Data type | Constraint |
| 1 | BookRackId | INT | PK, IDENTITY |
| 2 | BookRackName | NVARCHAR(MAX) | NULL |
| 3 | ShelfId | INT | FK(Shelf) |

**Entity: BookReview**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl No: | Data Name | Data type | Constraint |
| 1 | BookReviewId | INT | PK, IDENTITY |
| 2 | BookId | INT | FK(Book) |
| 3 | UserId | NVARCHAR(MAX) | NULL |
| 6 | Comments | NVARCHAR(MAX) | NULL |
| 7 | IsActive | BIT | NOT NULL |

**Entity:BookWishlist**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl No: | Data Name | Data type | Constraint |
| 1 | BookWishlistId | INT | PK, IDENTITY |
| 2 | IsActive | INT | FK (USER) |
|  | UserId | NVARCHAR(MAX) | NOT NULL |
| 3 | BookId | INT | FK(BOOK) |

**Entity: BorrowedBook**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl No: | Data Name | Data type | Constraint |
| 1 | BorrowedBookId | INT | PK, IDENTITY |
| 2 | UserId | NVARCHAR (MAX) | NULL |
| 3 | BookId | INT | FK(Book) |
|  | BookCopyId | INT | FK(BookCopyId) |
| 4 | RequestTimestamp | DATETIME | NULL |
| 5 | BorrowDate | DATETIME | NULL |
|  | DueDate | DATETIME | NULL |
|  | Status | NVARCHAR(MAX) | NULL |
|  | IsActive | BIT | NULL |
|  | ActualReturnDate | DATETIME | NULL |
|  | Comment | NVARCHAR(MAX) | NULL |

**Entity: Invoice**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl No: | Data Name | Data type | Constraint |
| 1 | InvoiceId | INT | PK, IDENTITY |
| 2 | BorrowId | INT | FK(BorrowId) |
| 3 | UserId | NVARCHAR (MAX) | NULL |
| 4 | Payment | DECIMAL(10.2) | NULL |
| 5 | Refund | DECIMAL(10.2) | NULL |
| 6 | TransactionDate | DATETIME | NULL |
| 7 | Fine | DECIMAL(10.2) | NULL |
| 8 | MiscellaneousFines | DECIMAL(10.2) | NULL |
| 9 | Remarks | NVARCHAR (MAX) | NULL |
| 10 | TransactionId | NVARCHAR (MAX) | NULL |
| 11 | GenerateTransactionId | NVARCHAR (MAX) | NOT NULL |

**Entity:** **Shelf**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl No: | Data Name | Data type | Constraint |
| 1 | ShelfId | INT | PK, IDENTITY |
| 2 | IsActive | BIT | NOT NULL |
| 3 | ShelfName | NVARCHAR(MAX) | NULL |
| 4 | BookFloorId | INT | FK(BookFloor) |

**Entity: Subcategory**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl No: | Data Name | Data type | Constraint |
| 1 | SubcategoryId | INT | PK, IDENTITY |
| 2 | DDCCode | NVARCHAR (MAX) | NOT NULL |
| 3 | Name | NVARCHAR(MAX) | NOT NULL |
| 4 | IsActive | BIT | NOT NULL |
| 5 | CategoryId | INT | FP(Category) |

**Entity: SubscriptionPlan**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl No: | Data Name | Data type | Constraint |
| 1 | SubscriptionPlanId | INT | PK, IDENTITY |
| 2 | PlanName | NVARCHAR(MAX) | NOT NULL |
| 3 | PlanDescription | NVARCHAR(MAX) | NOT NULL |
| 4 | PlanPrice | DECIMAL(18,2) | NULL |
| 5 | IsActive | BIT | NOT NULL |
| 6 | MonthlyFee | DECIMAL(10,2) | NULL |
| 7 | MaxAllowedBookPrice | DECIMAL(10,2) | NULL |

**Entity: SubscriptionUser**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl No: | Data Name | Data type | Constraint |
| 1 | SubscriptonUserId | INT | PK, IDENTITY |
| 2 | ApplicationUser | ApplicationUser | NULL |
| 3 | SubscriptionPlanId | INT | NOT NULL |
| 4 | TransactionId | NVARCHAR(MAX) | NULL |
| 5 | Accepted | BIT | NOT NULL |

**Entity: UserPreference**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl No: | Data Name | Data type | Constraint |
| 1 | UserPreferenceId | INT | PK, IDENTITY |
| 2 | IsActive | BIT | NOT NULL |
| 3 | UserInfo | NVARCHAR(MAX) | NULL |
| 4 | CategoryId | INT | FP(Category) |
| 5 | AuthorId | INT | FP(Author) |

Methodology

Software development life cycle provides the overview and procedures to develop software. For our “Online Bookstore Management”, the life cycle is given below-At first stage of SDLC we have to measure feasibility study of our proposed system that is system feasibility study of economics, implementation etc.



**Requirements Analysis:** The existing system is evaluated. Deficiencies are identified. This can be done by interviewing users of the system and consulting with support personnel. The new system requirements are defined. In particular, the deficiencies in the existing system must be addressed with specific proposals for improvement. Other factors defined include needed features, functions.

**Design:** The proposed system is designed. Plans are laid out concerning the physical construction, hardware, operating systems, programming, communications and security issues.  
**Development:** The new system is developed. The new components and programs must be obtained and installed. Users of the system must be trained in its use.

**Testing:** All aspects of performance must be tested. If necessary, adjustments must be made at this stage. Tests performed by quality assurance (QA) teams may include systems integration and system testing.

**Maintenance:**

This step involves changing and updating the system once it is in place. Hardware or software may need to be upgraded, replaced or changed in some way to better fit the needs of the end-users continuously. Users of the system should be kept up-to-date concerning the latest modifications and procedures.

Software Requirements

**TECHNOLOGY USED**

**Software and Operating System:**

* Microsoft Windows 10 Pro
* Microsoft Visual Studio 2017
* Microsoft SQL Server 2016
* Microsoft Office 2016
* Visual Studio Code

**Technologies:**

* Restful API (MVC API & ASP.NET CORE)
* HTML5, CSS3, JavaScript
* TypeScript
* Bootstrap 4.4.1
* Entity Framework 6
* Angular 6
* jQuery
* Some third-party add-ins

Use Case Diagram

### 

Class Diagram

### 

Code Samples

### Model Classes:

### 

### Purchase Repository:

### 

### 

### Cart Repository:

### IRepository:

### 

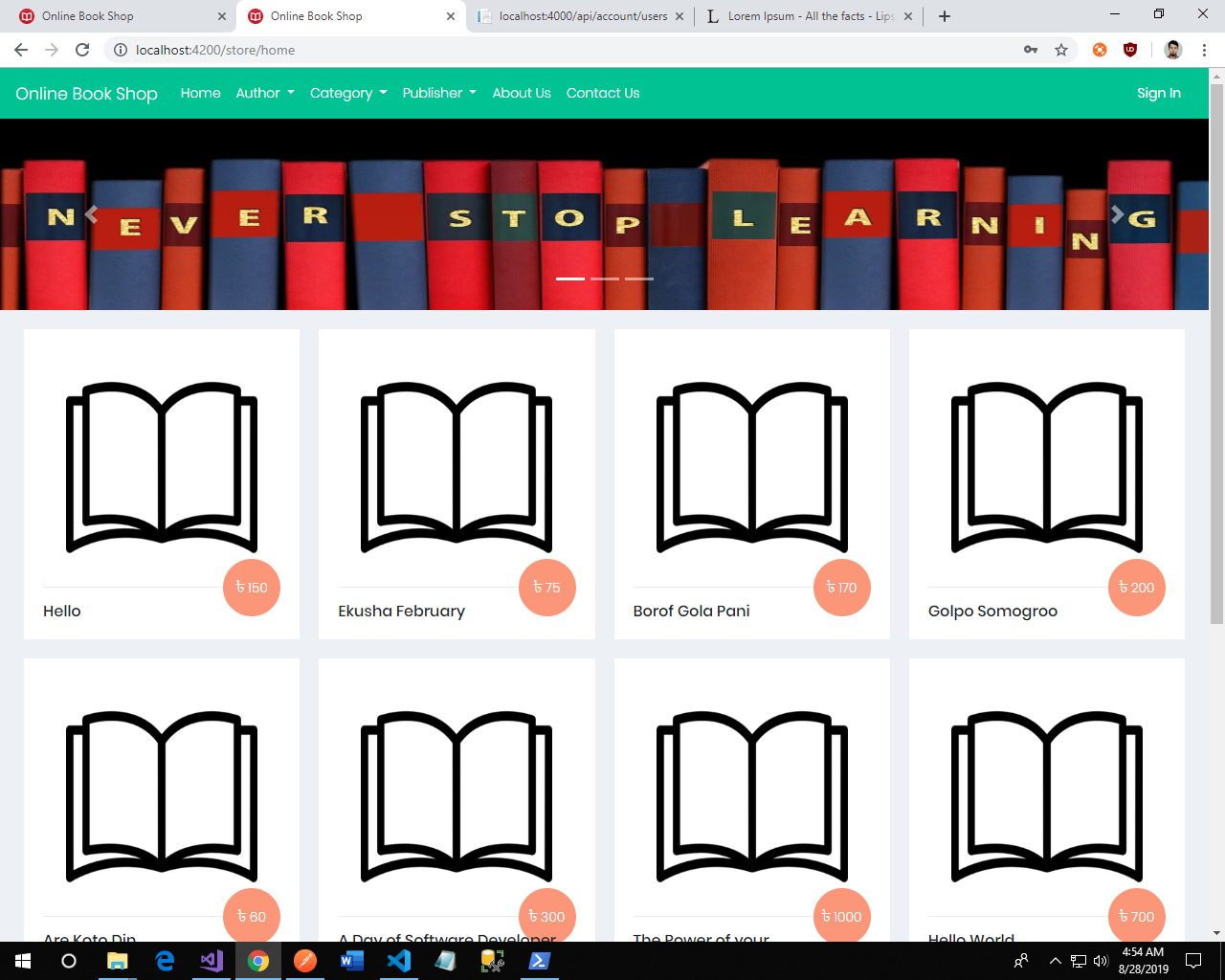
### Book Controller:

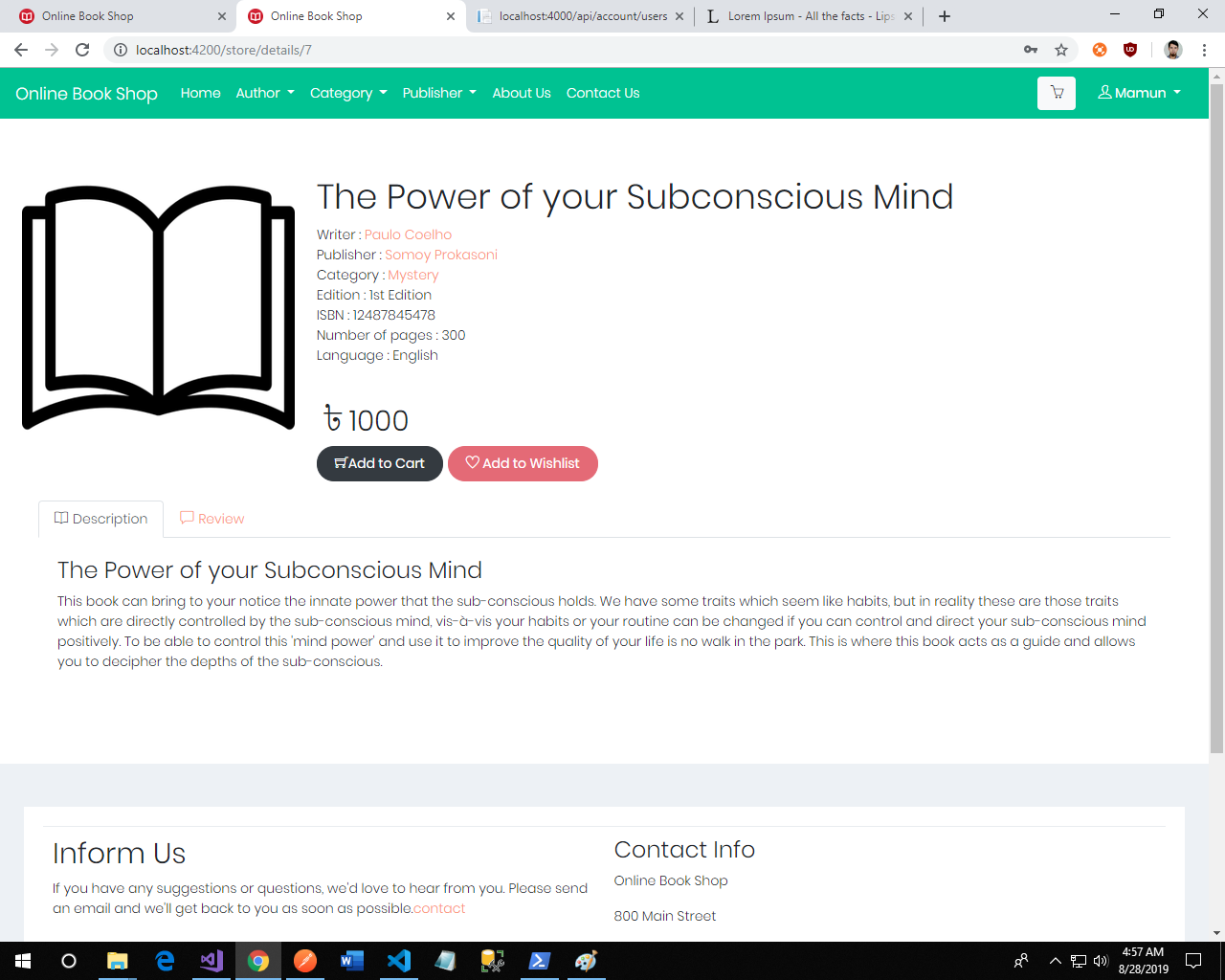
### 

### Cart Controller:

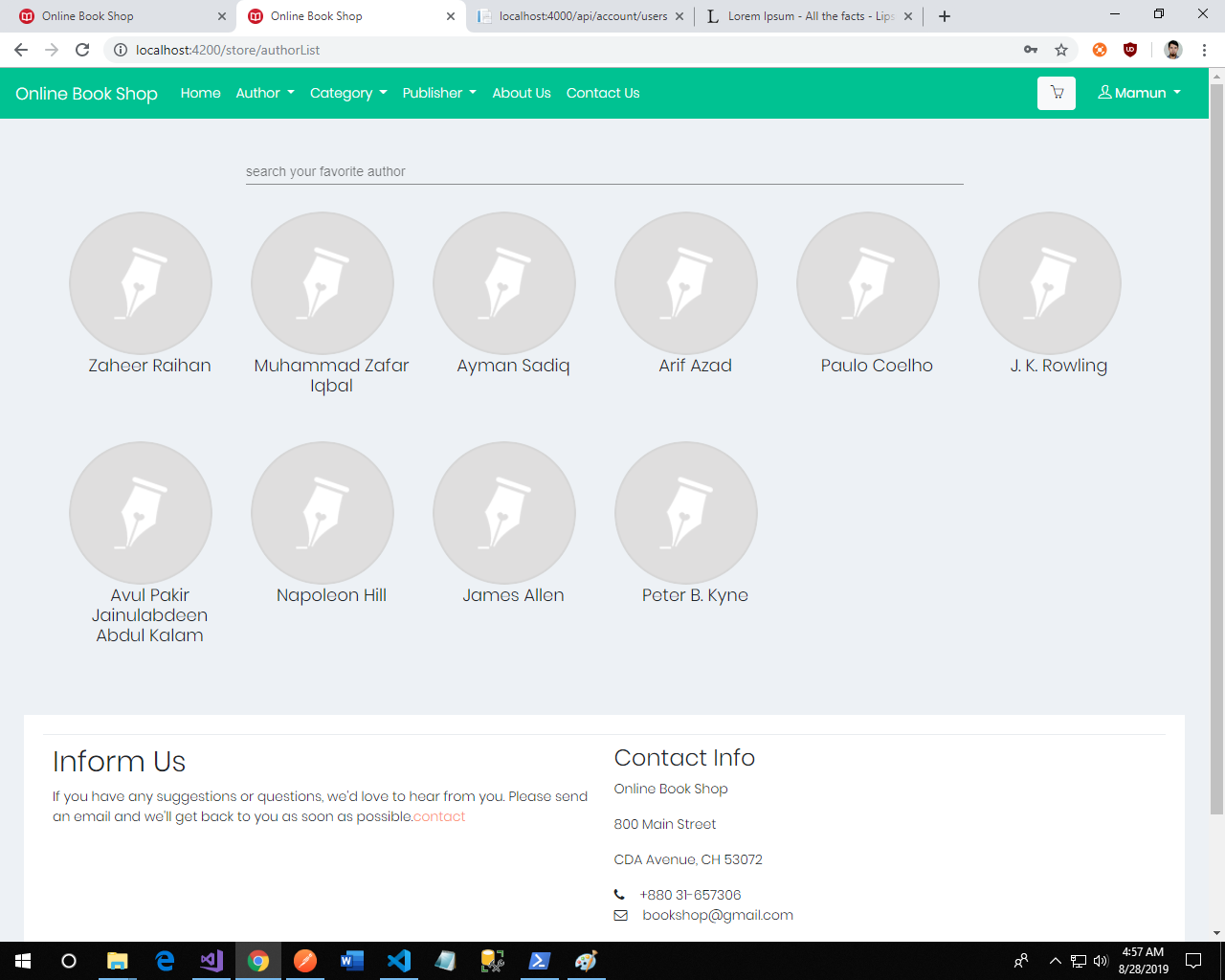
### 

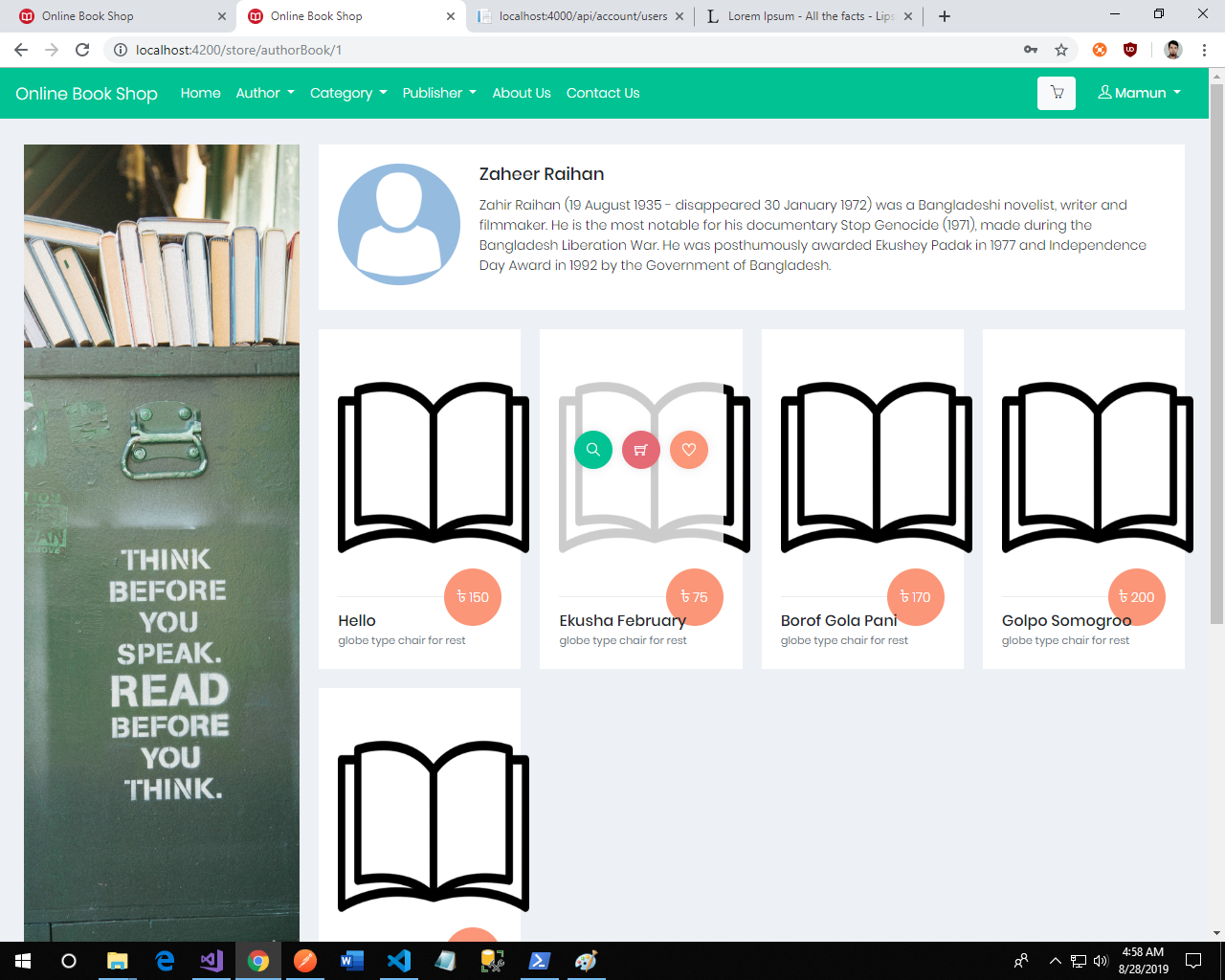
Bookstore Interface

**Landing Page:** In this project user visit this site for buying their required book. They search book by author, category, publication.

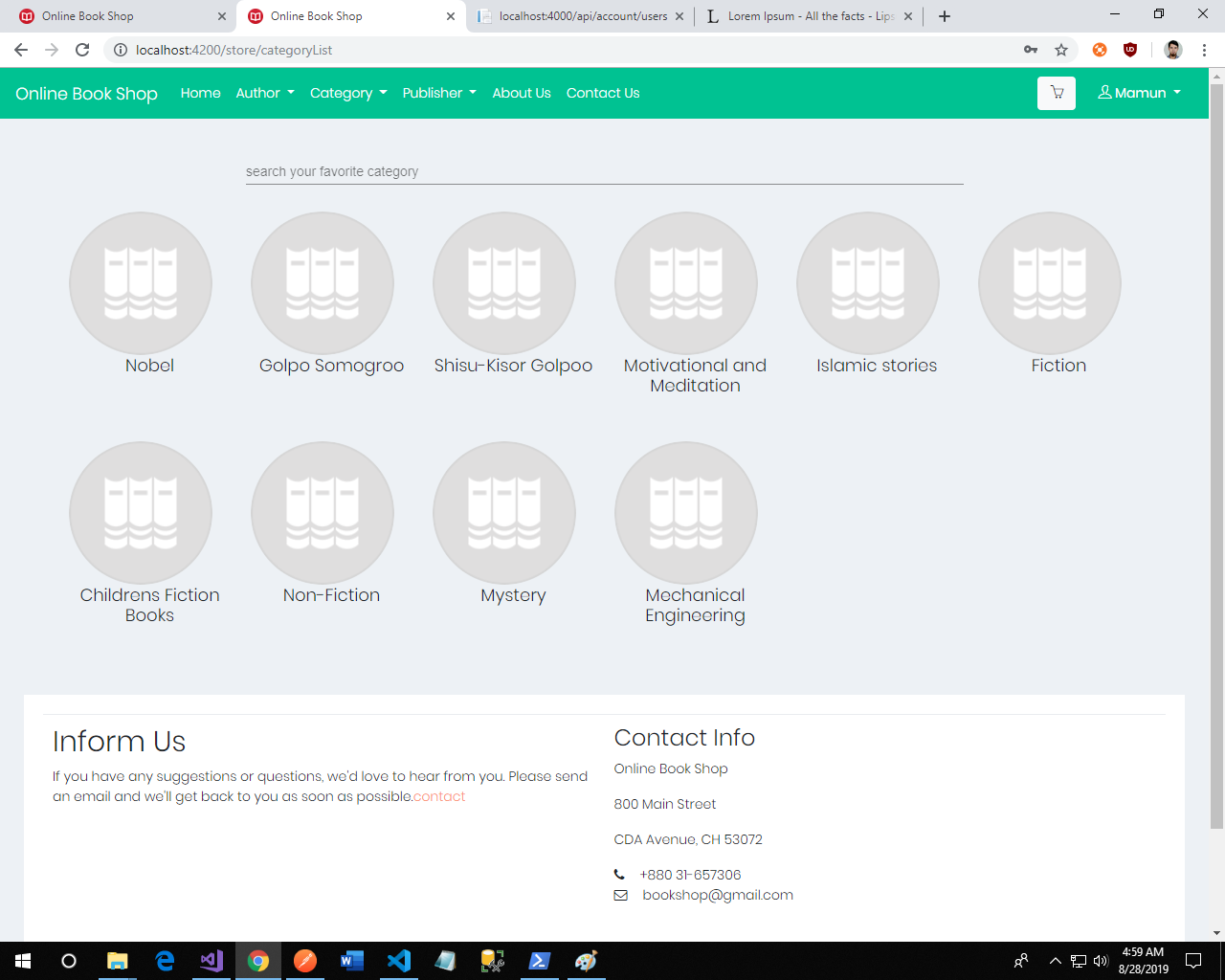
**Book Details:** If user want to know about the book, they can see the book details. 

**Authors:** User can get the information of specific author and his writing books.

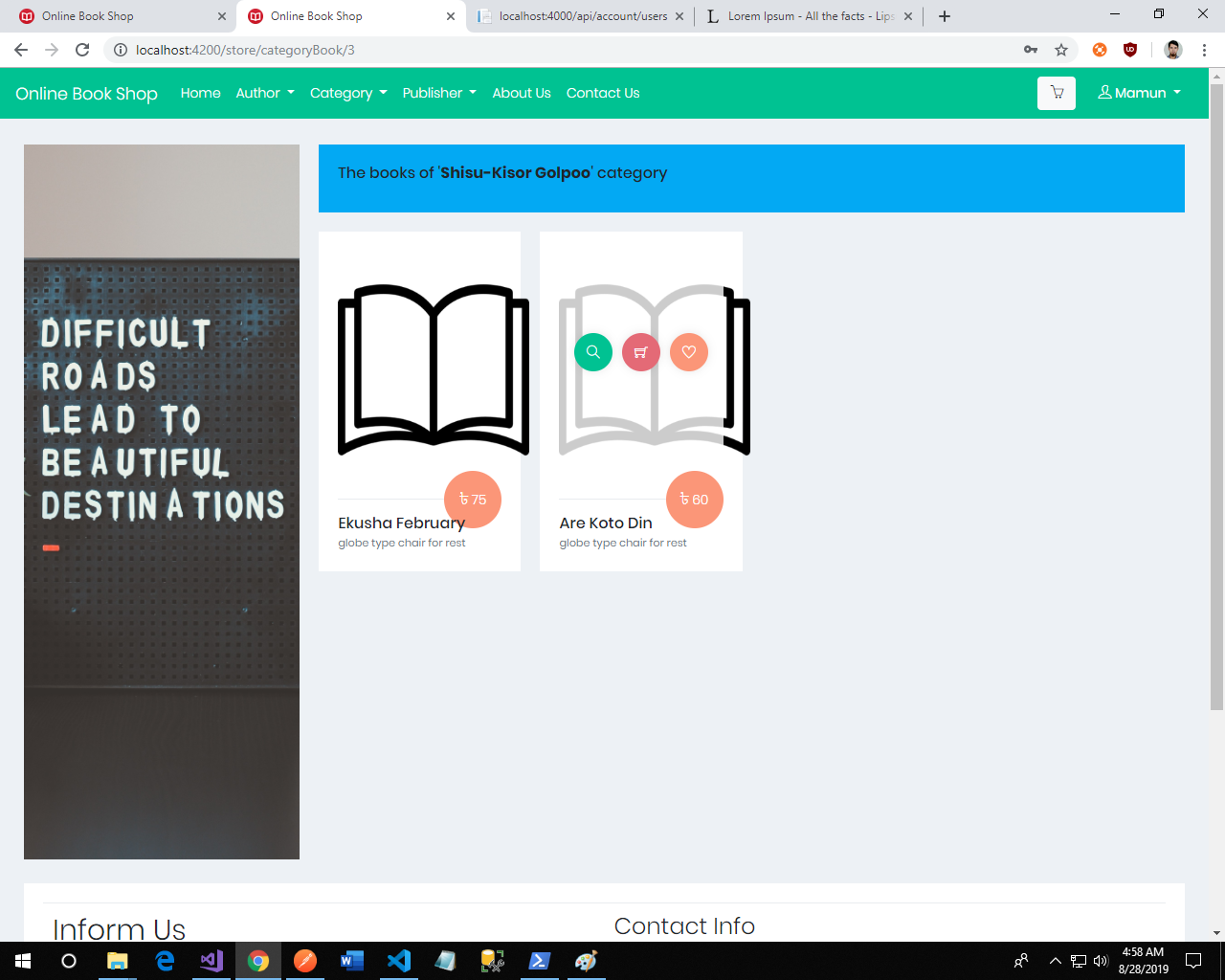


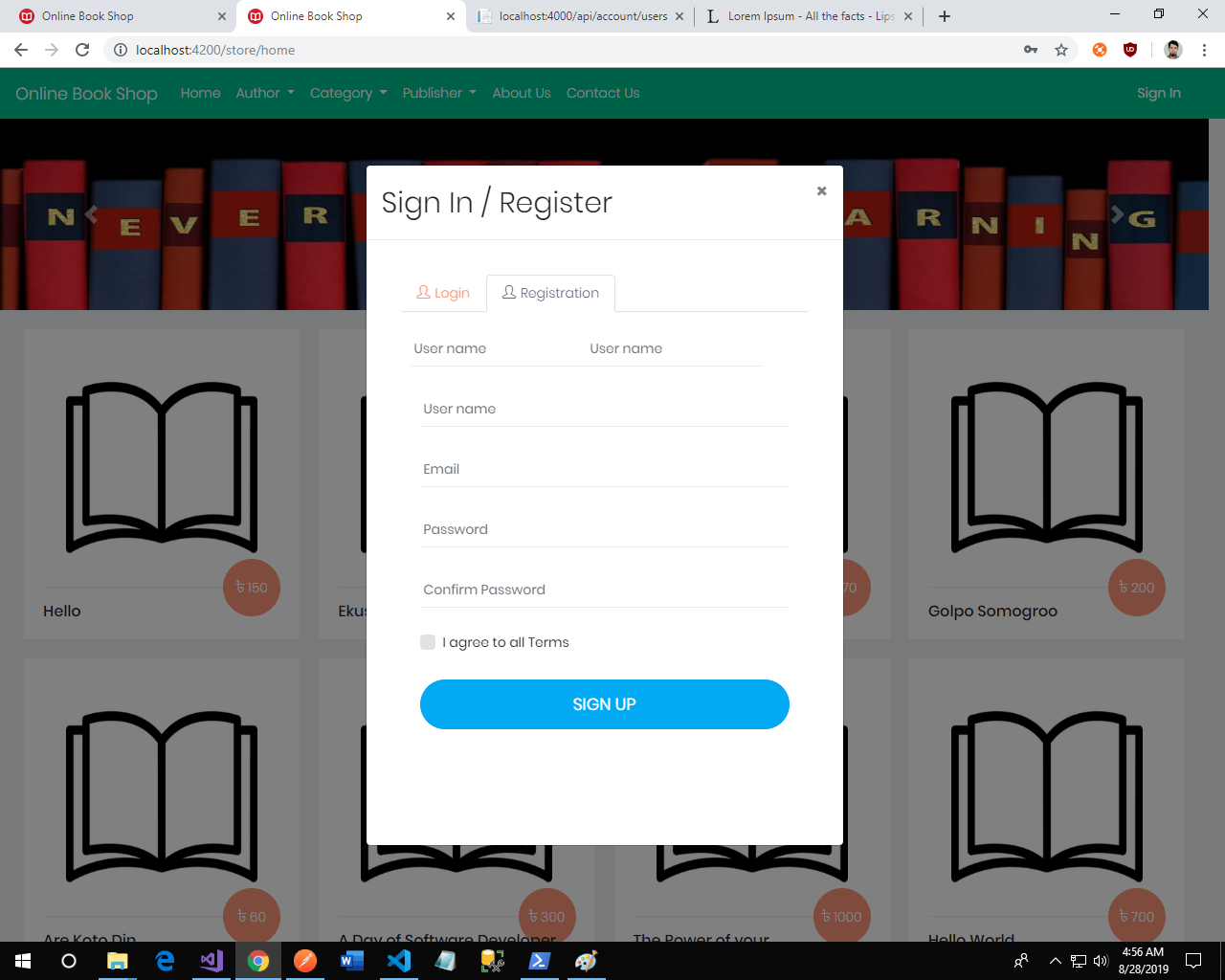
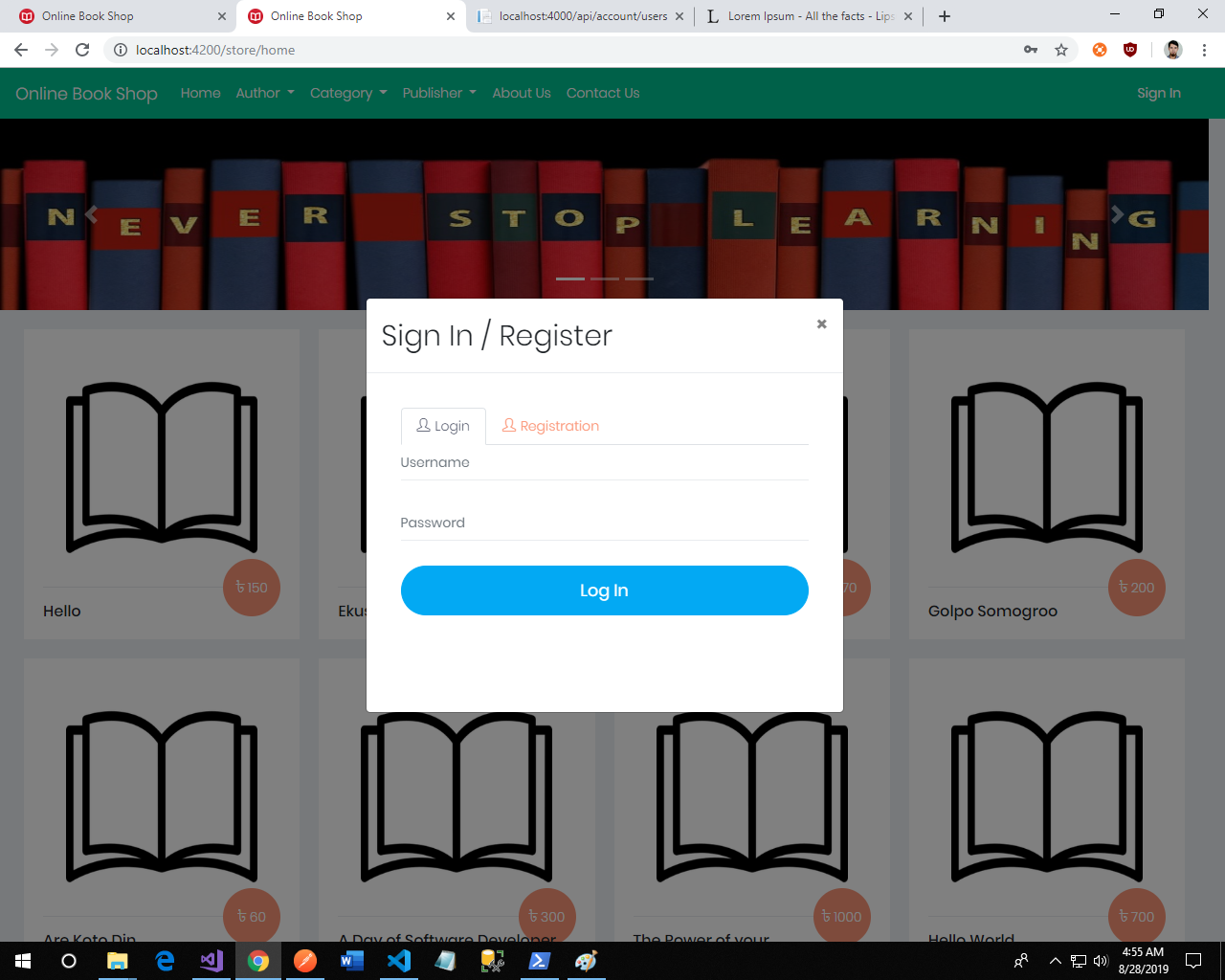
**Author wise books:**

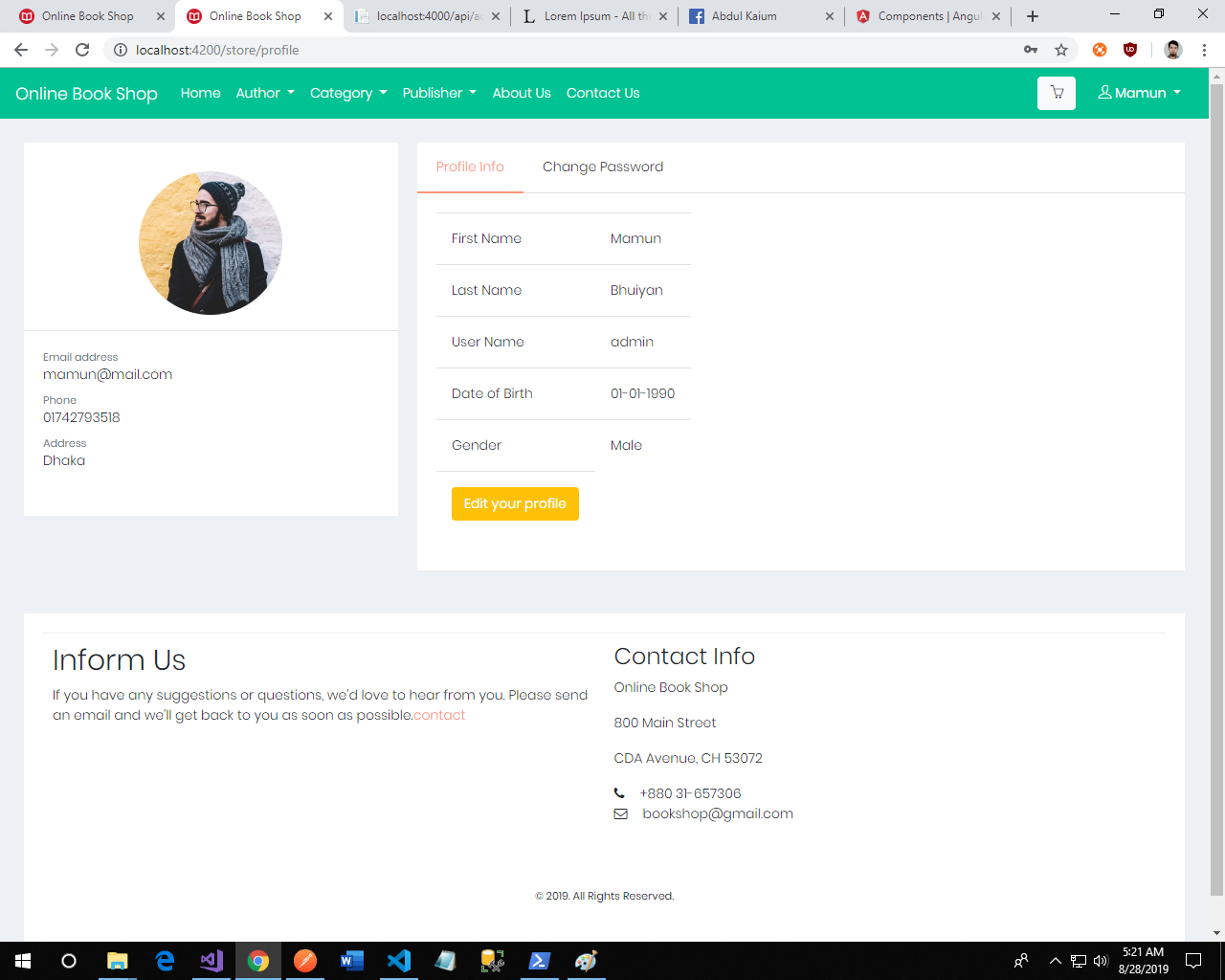
**Category list:**

****

**Category wise books:**

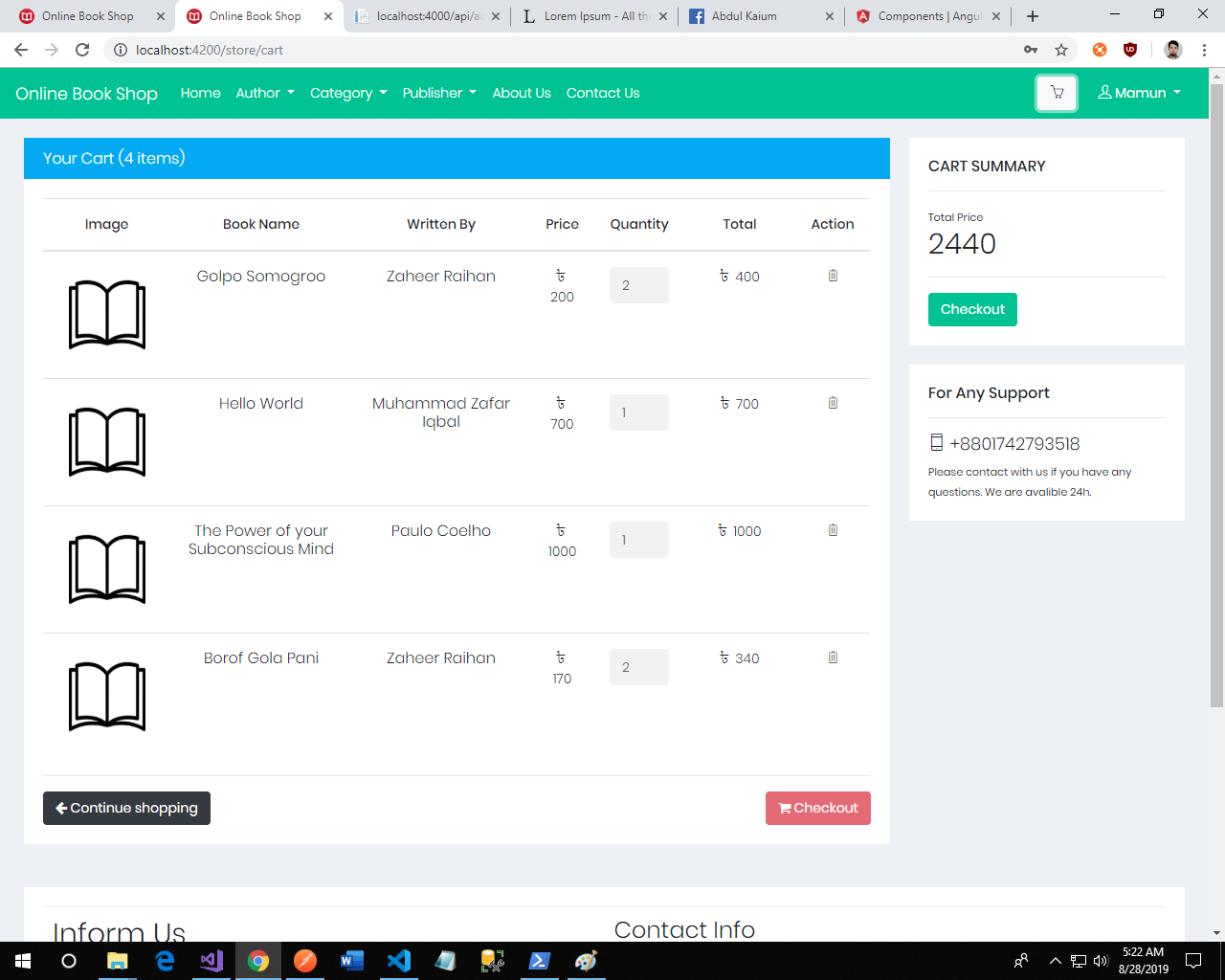
****

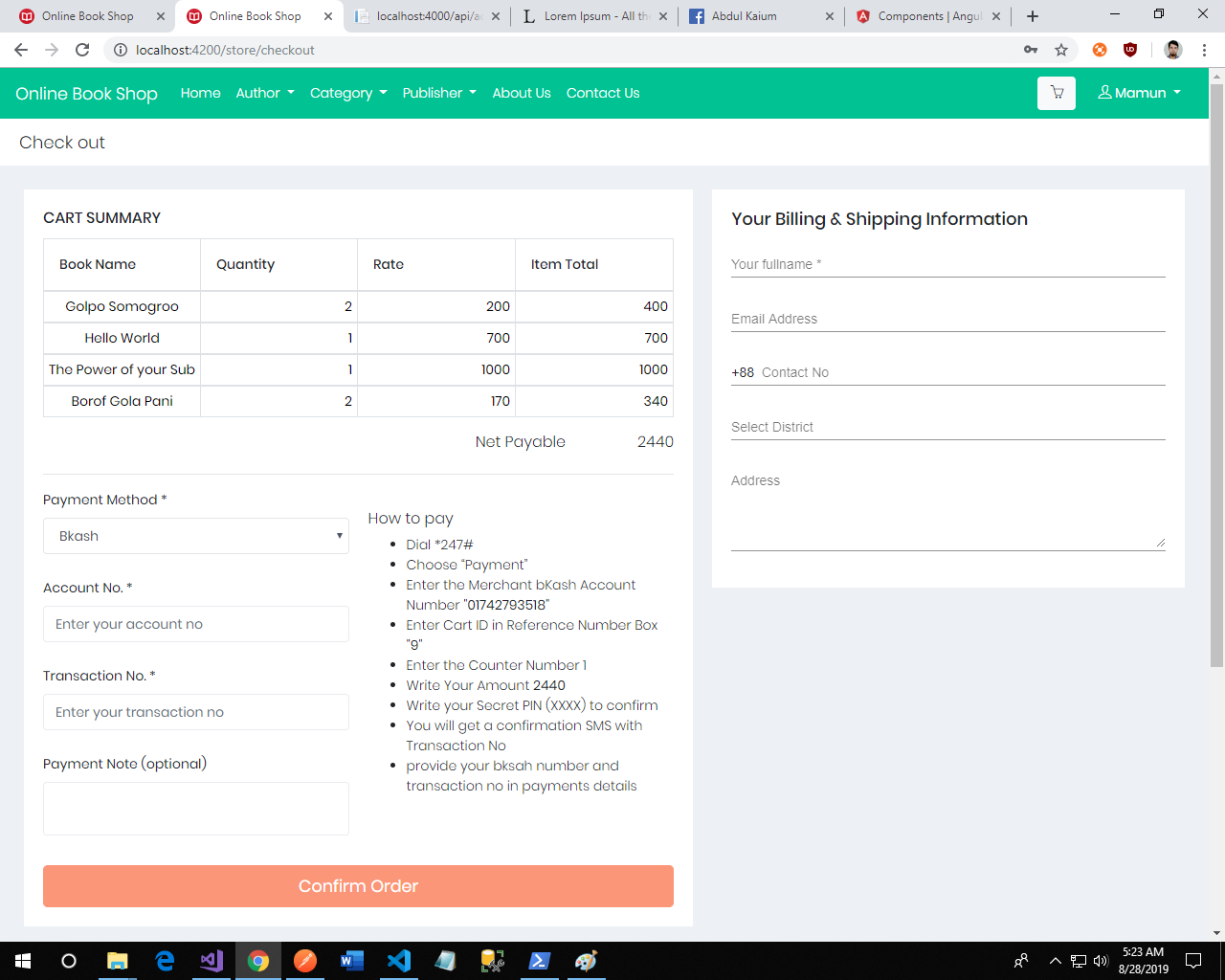
**Register & Login:** In log in page user log in by user name and password if he or she registered. If he or she didn’t registered user then go to the register page.

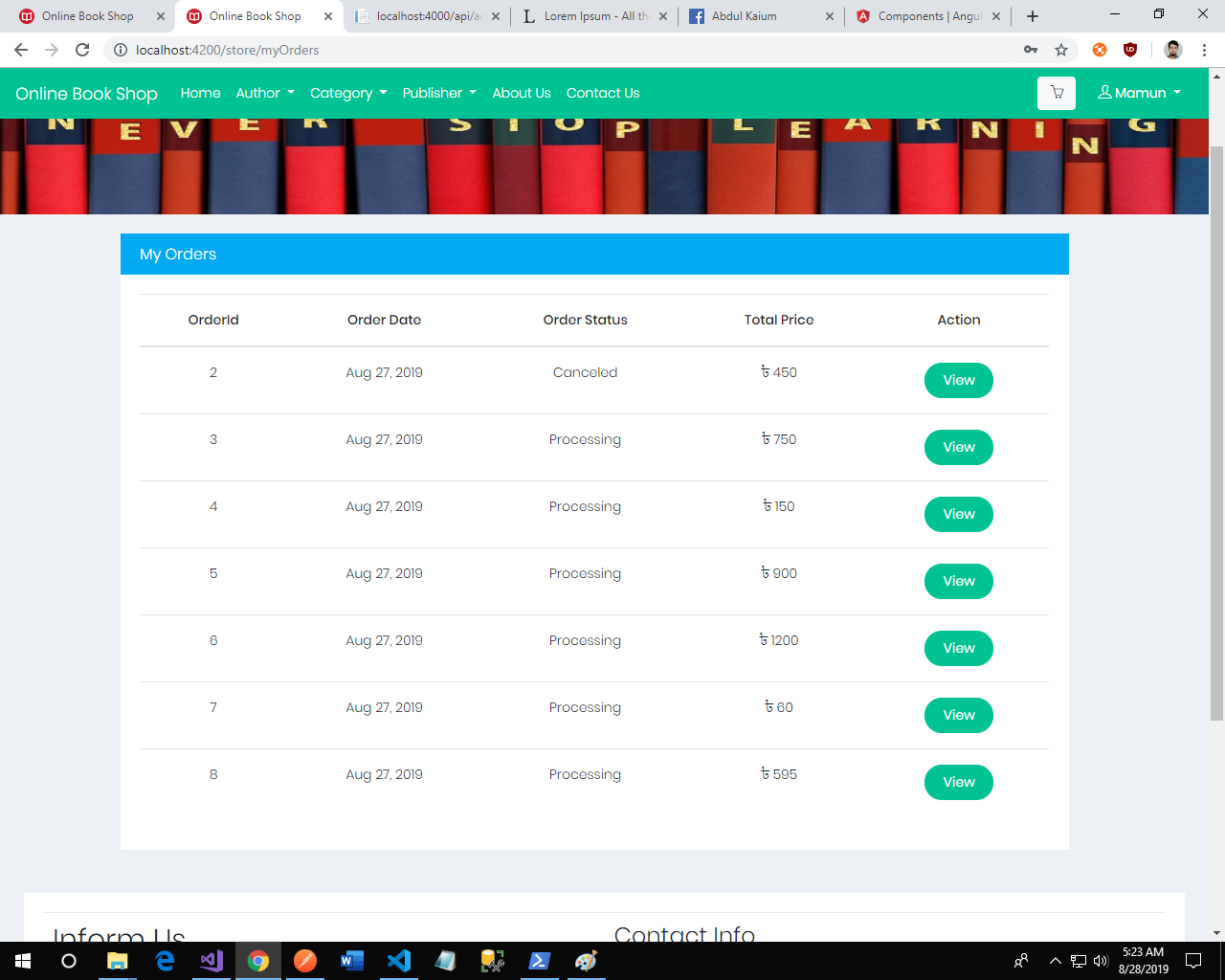
**User Profile :**  A logged in user can view and update his/her profile and also change the password.

**Wishlist and Cart: Authenticate user can add any book to their Wishlist and cart from home page, author wise book page, category wise book page or publisher wise book page.**



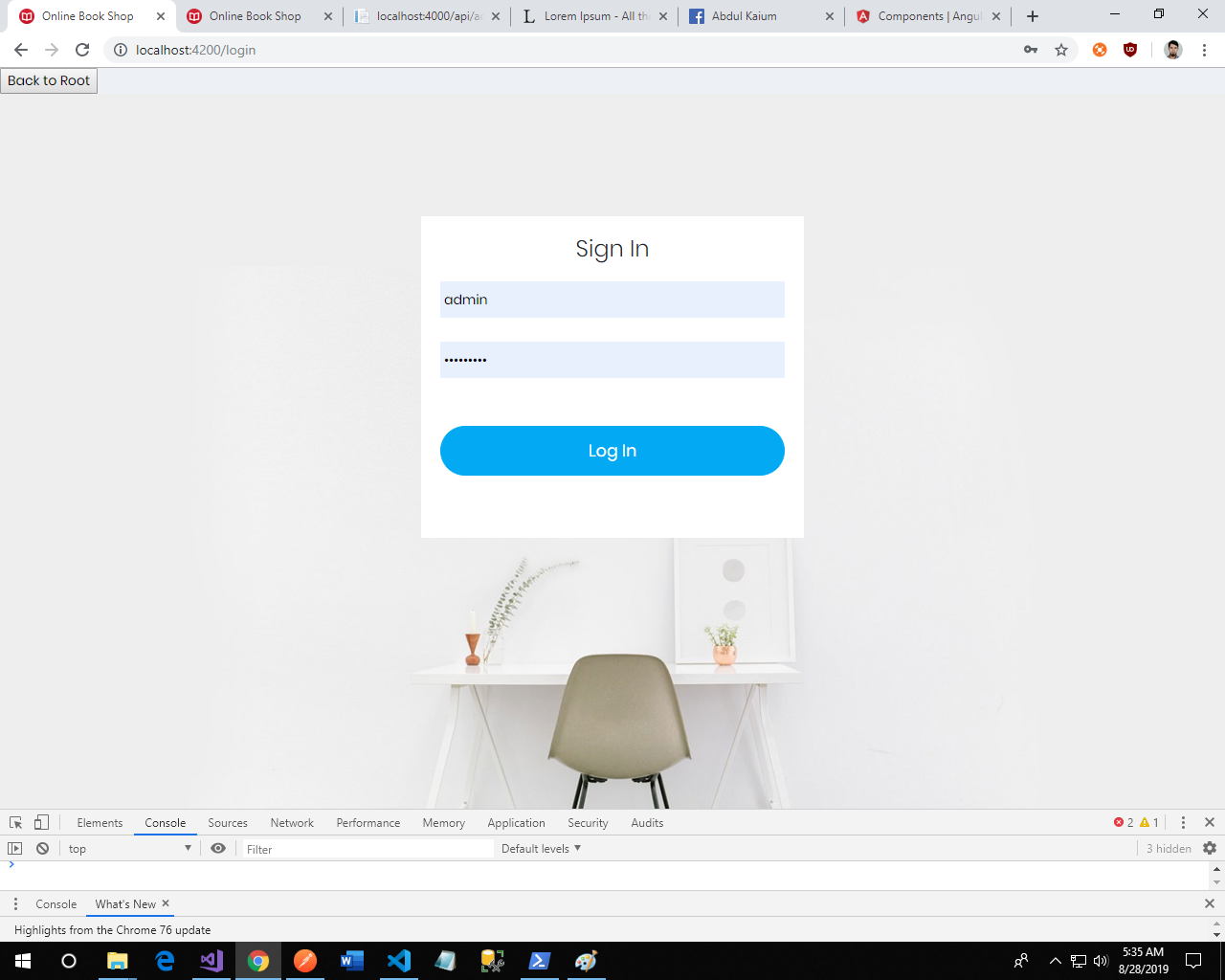


**Checkout: To order books user need to go to checkout page. User can order book by giving shipping address and payment details.**

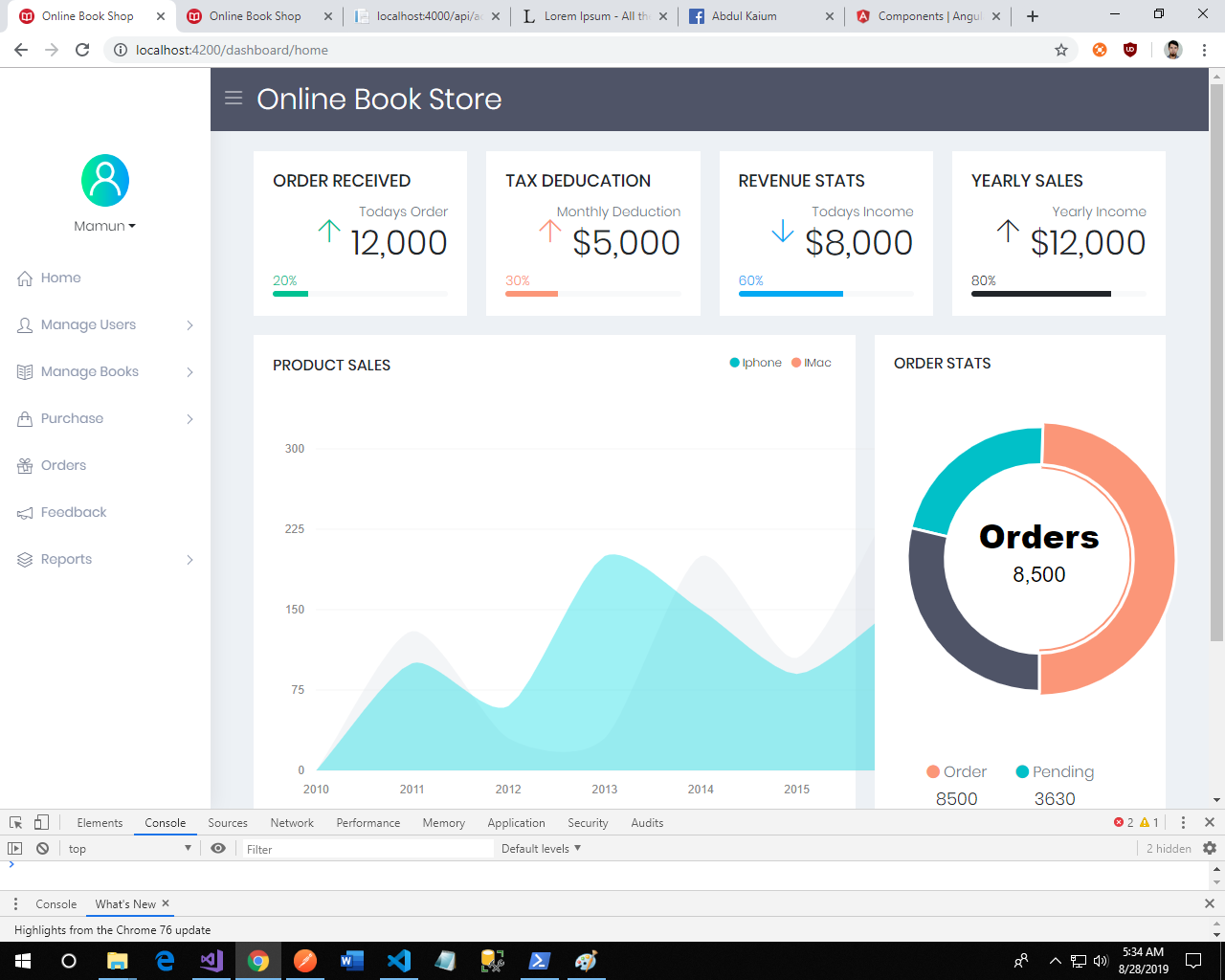
**Orderlist of user:** 

### Admin Dashboard Interface

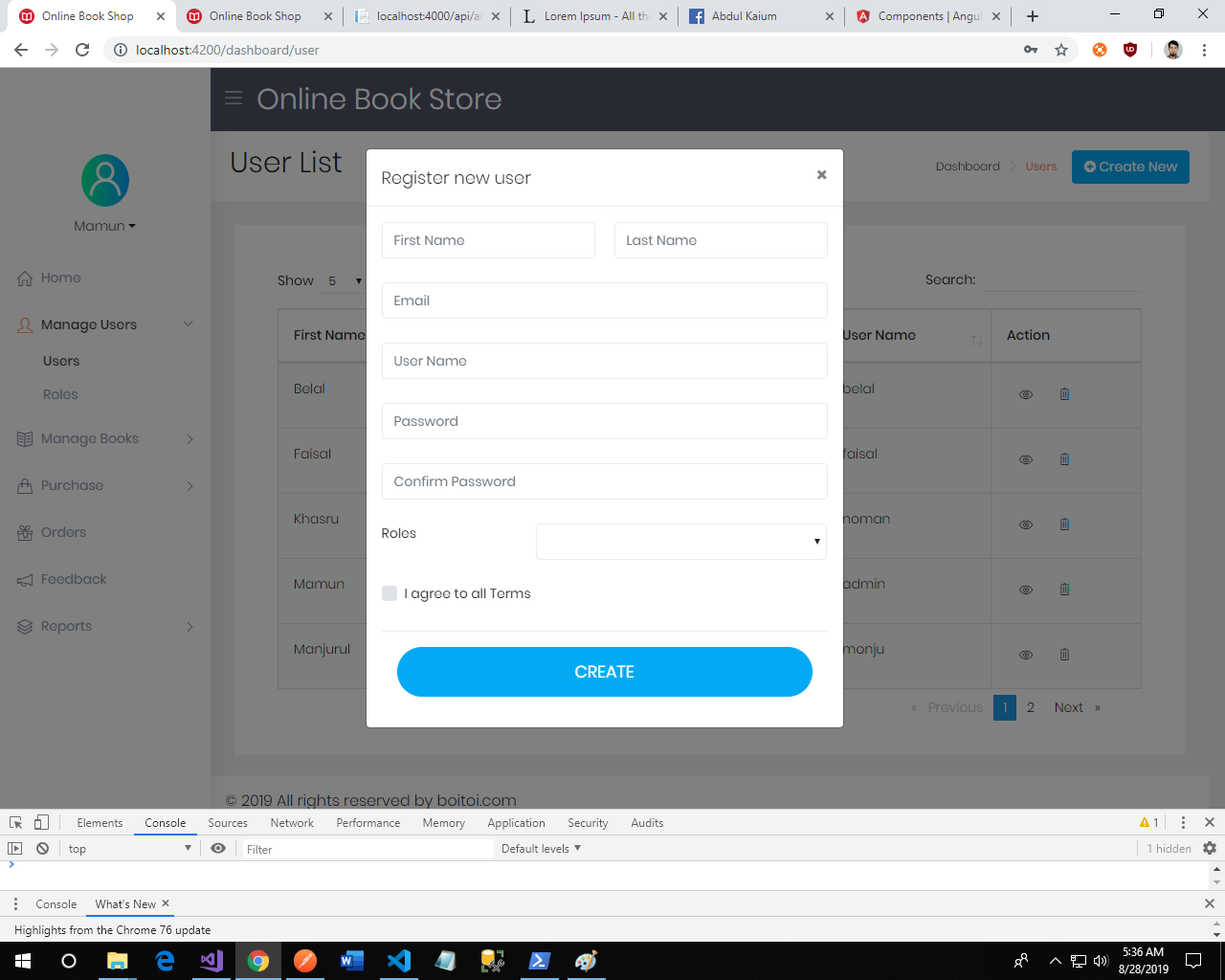
**Admin Login:**  Only Employees can login to the dashboard page , they can log in to store.

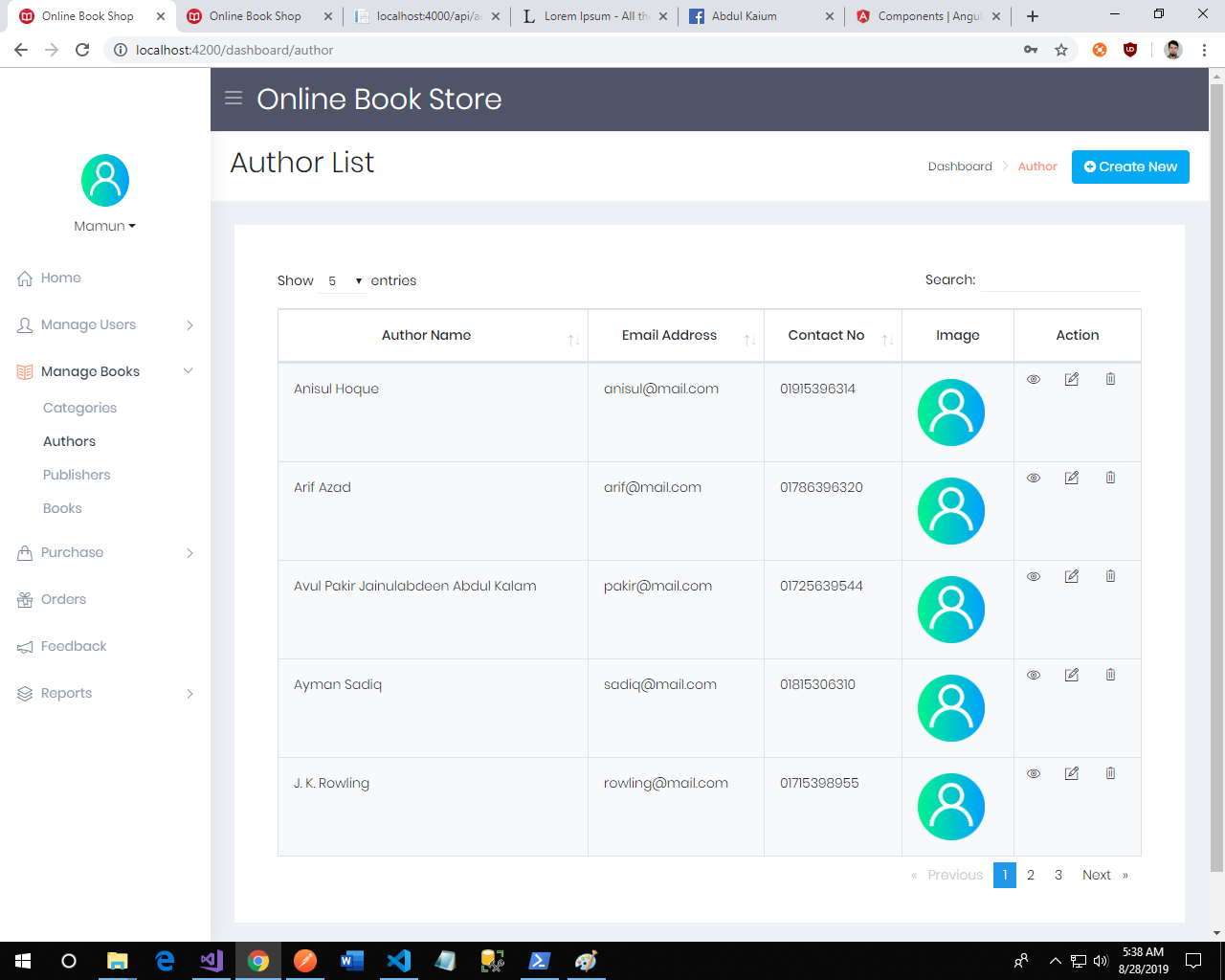


**Home Page :**  Dashboard home page are designed with some static components. But those could be very useful if we dynamically generate.

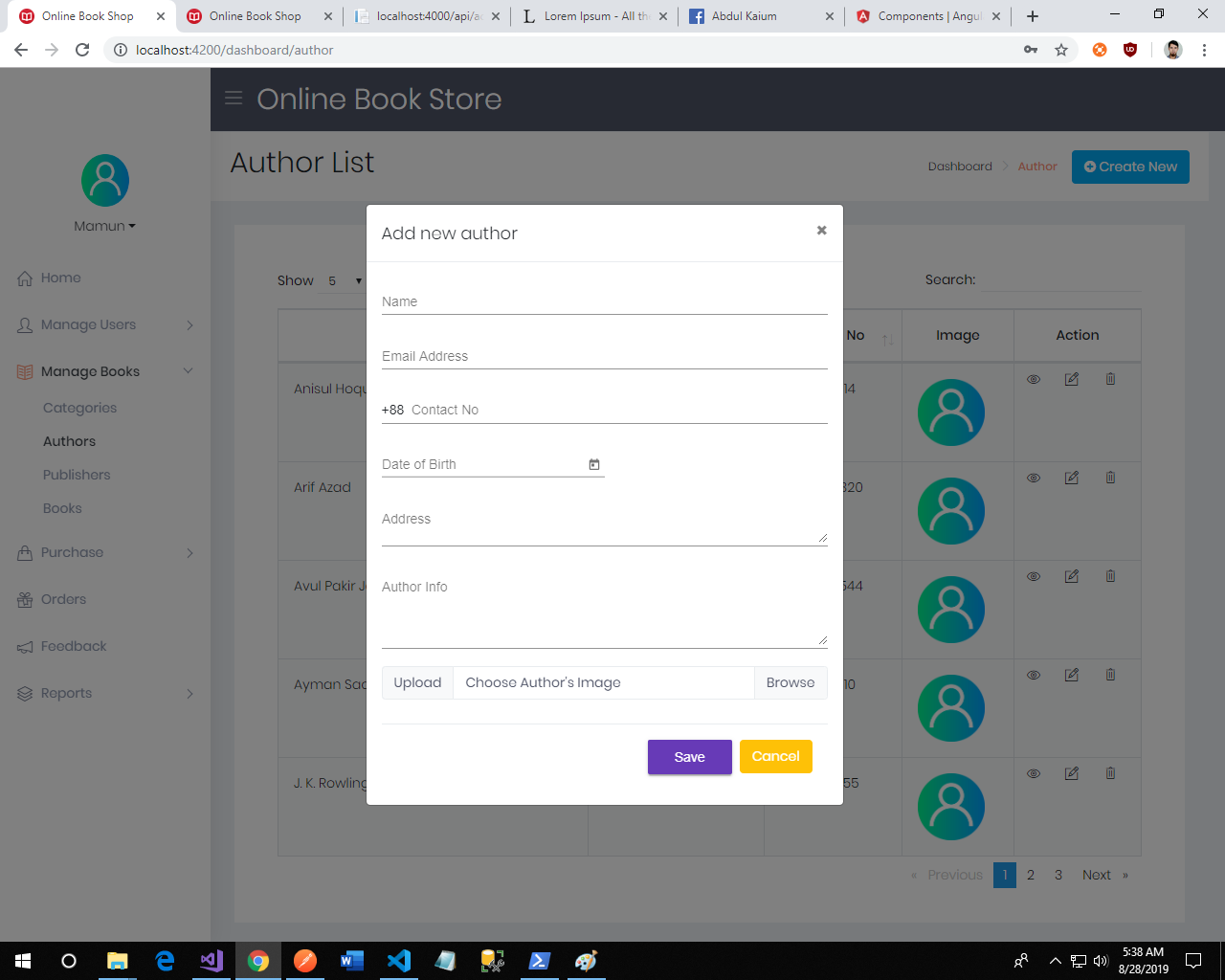


**Mange User:**  Admin can register new user and role for managing our site. He can also manage the user roles. He can view and delete the users and roles.

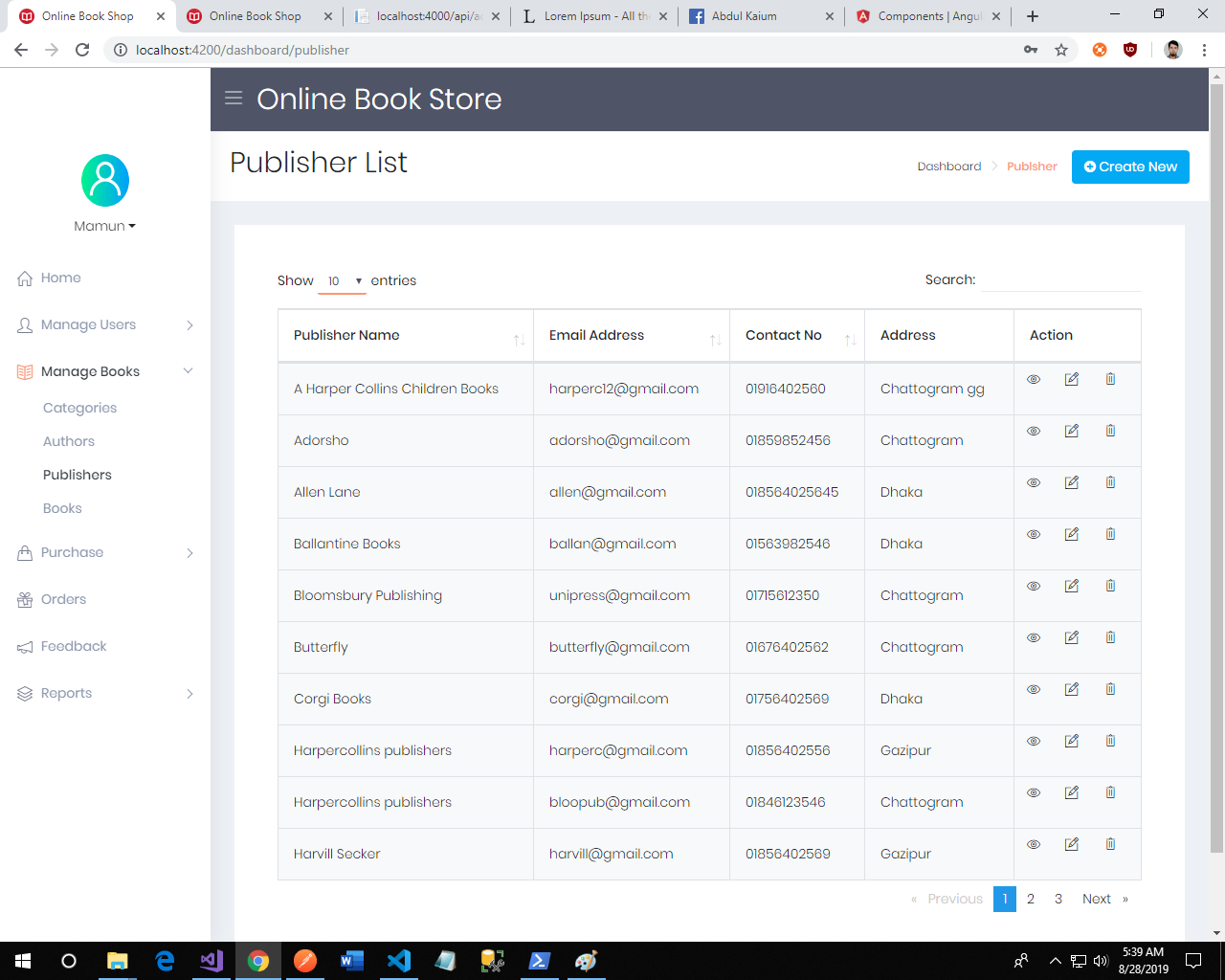


**Manage Books:** In order to manage book data, we create category, author and publisher. The authorized employee can create, view, update and delete those data. After that we can store our books. The book data created with opening stock and it can be updated and viewed. Here some screenshots of those operations.  
 *Author List*: 

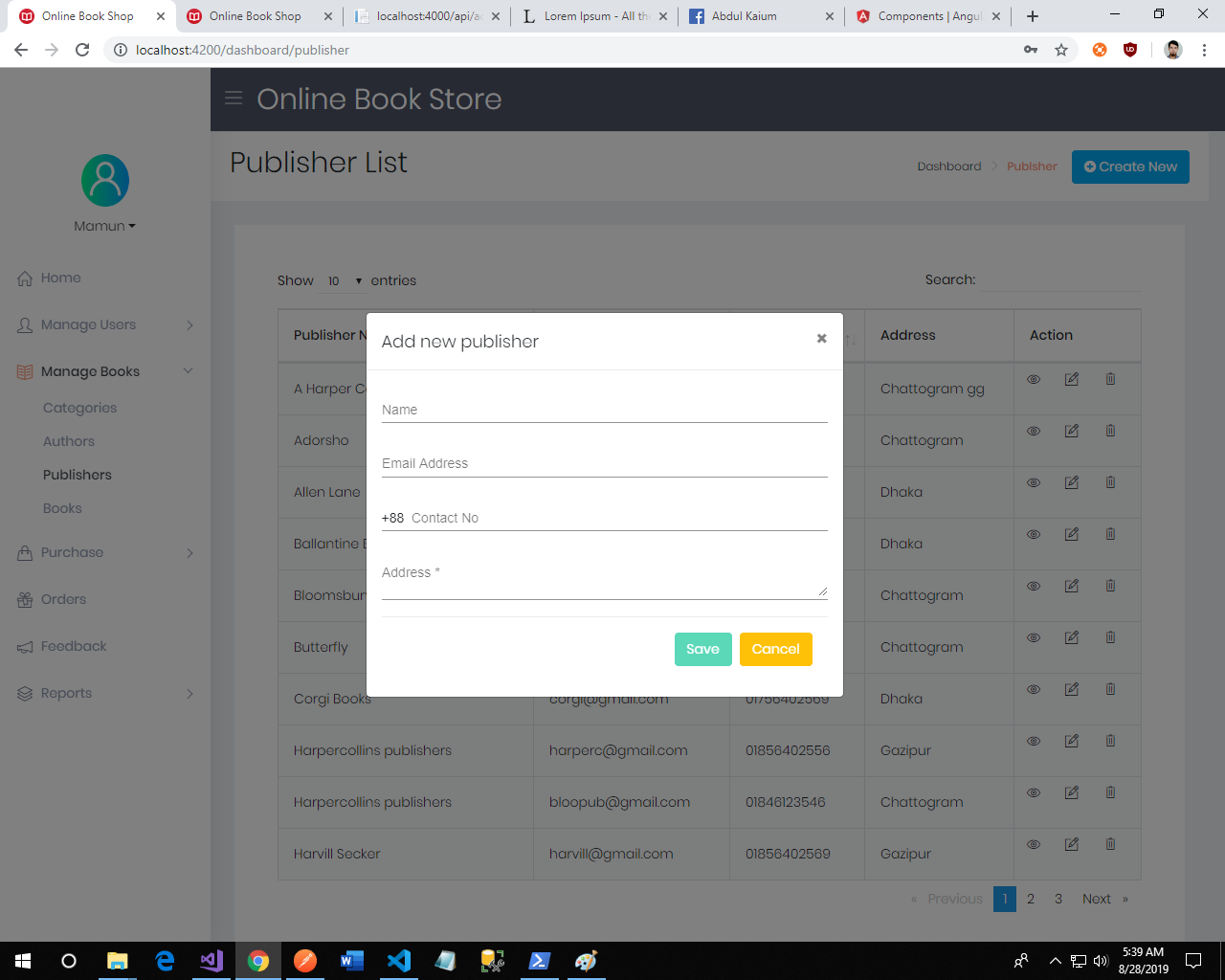
*Author Create Form:*

**

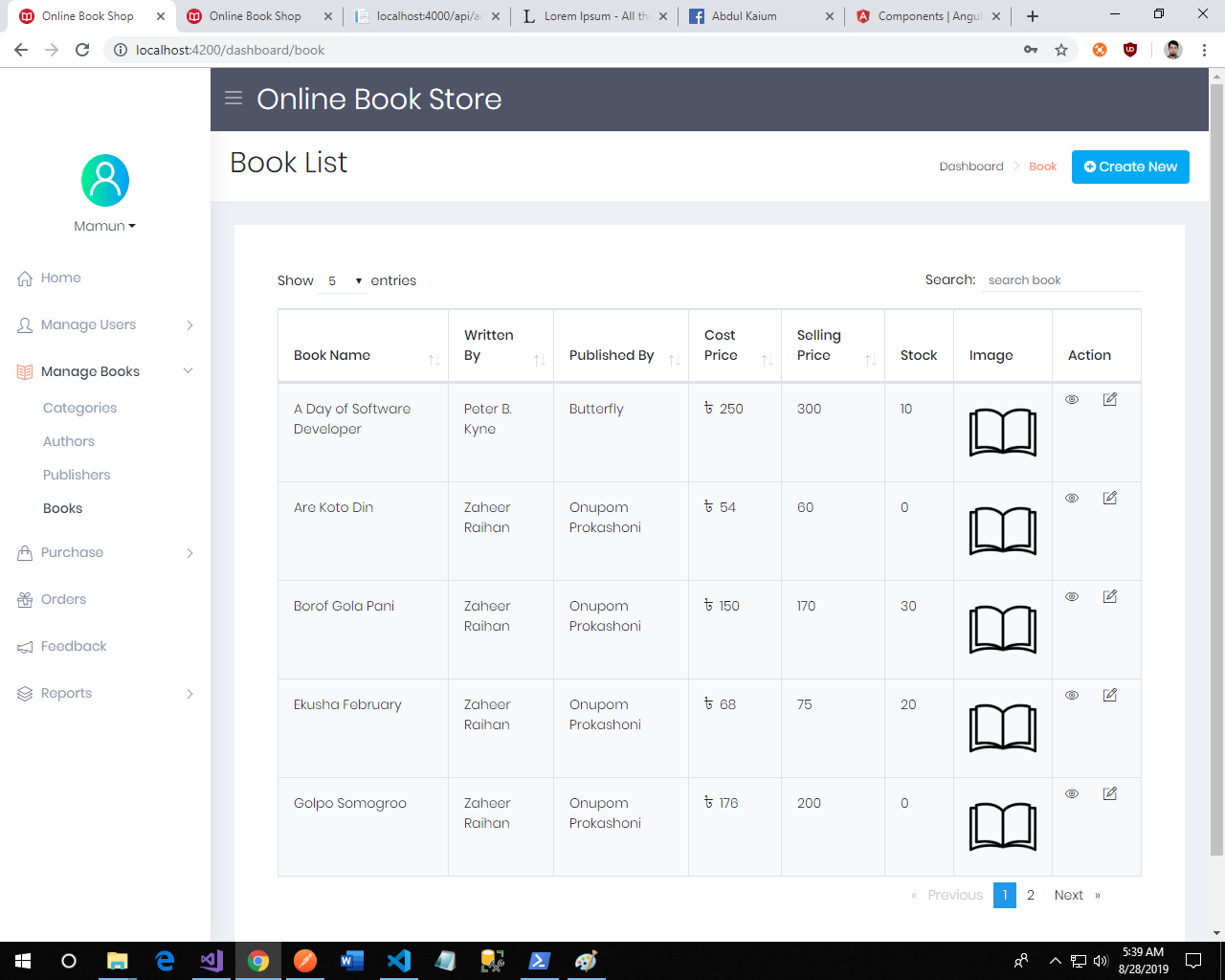
*Publisher list:*

**

*Publisher Create Form:*

**

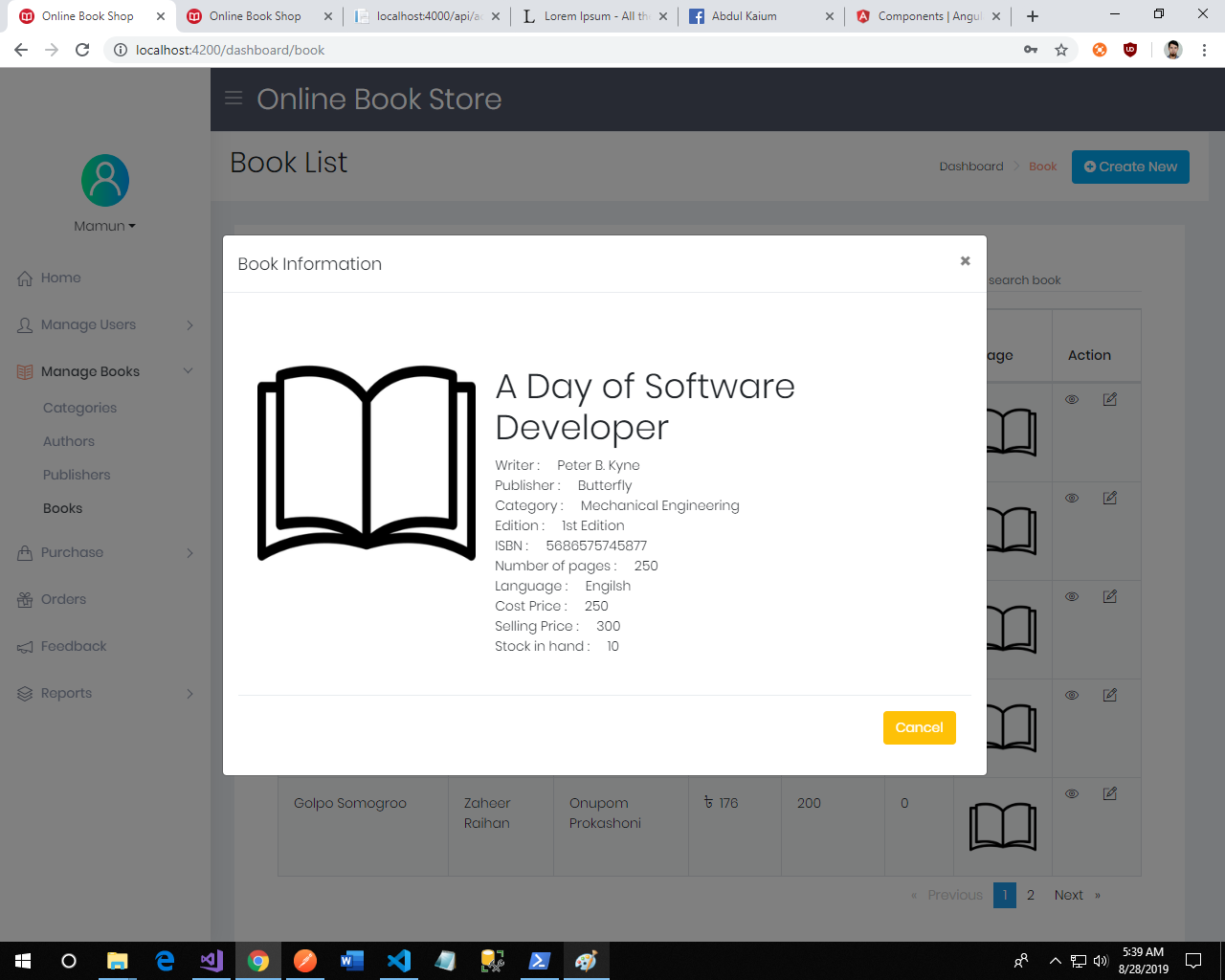
*Book list Form:*

**

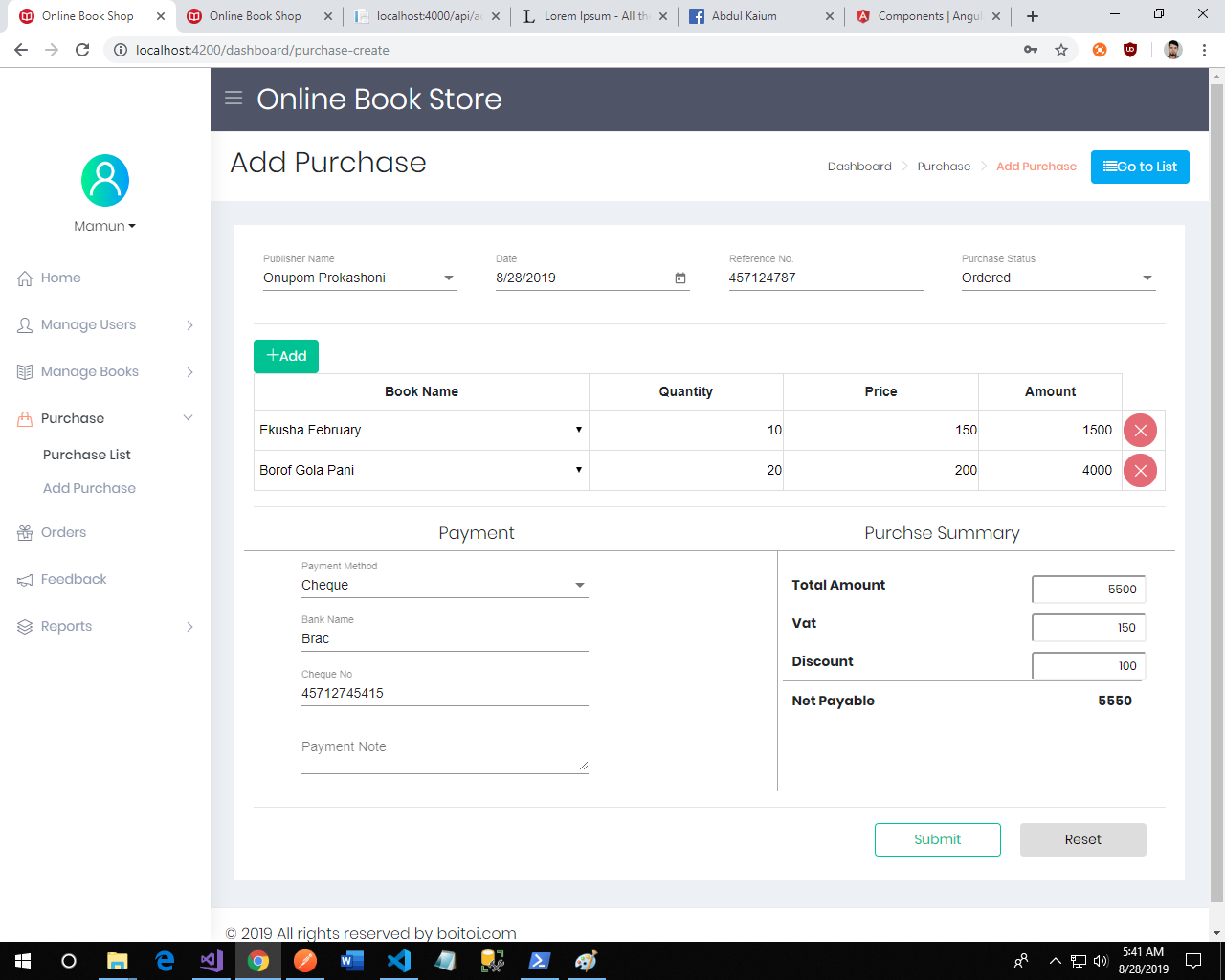
*Book Create Form:*

**

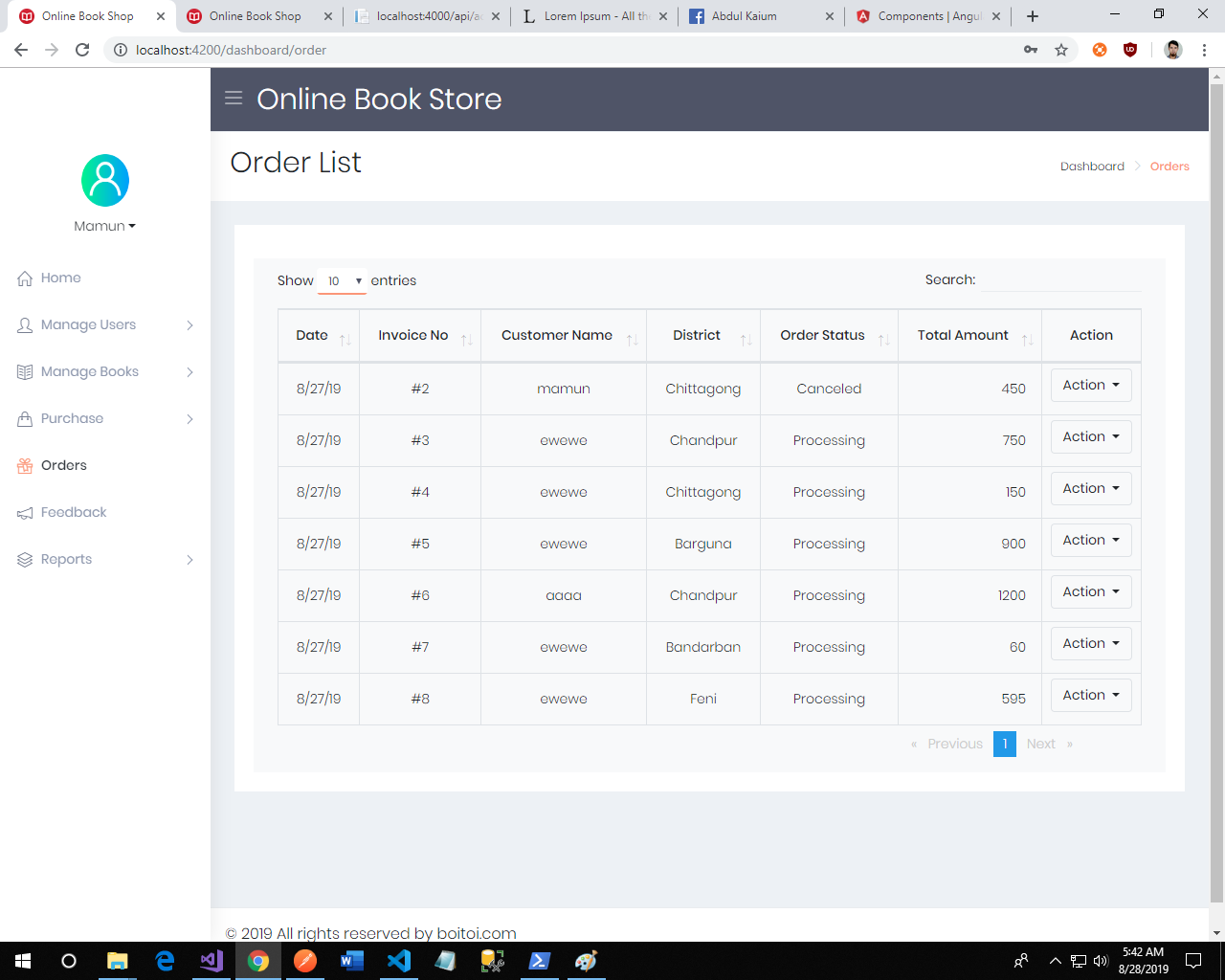
*Book View:*

**

**Purchase Books:** Authorized employee can entry the purchases of books. After purchase entry the respective book stock also updated. In order to entry purchase we also provide publisher and payment info. Some screenshots are given below.



**Order List**: The respective employee manages orders like view order details, print and update the order status.

*Order list:*

*Order details and print:*

**

CONCLUSION

This internet-based application is designed to cater to all levels of management by providing comprehensive organizational information. It serves as an effective information management system for the bookstore.

The bookstore management system's key functionality includes generating sales statistics for any given period, encompassing details such as book name, publisher, ISBN number, number of copies sold, and the corresponding sales revenue.

When a customer requests a book that is not currently available in the bookstore's inventory, the system prompts the customer to enter the full details of the requested book for procurement. If the requested book is in stock, the system displays the exact number of available copies and the rack number where the book is located. Conversely, if the book is not in stock, the system increments a request field for that book.

In the proposed system, once a customer selects a book for purchase, the sales clerk enters the ISBN number of the book. The bookstore management system then updates the stock and generates a sales receipt for the purchased book. This streamlined process ensures efficient book procurement, inventory management, and sales tracking for the bookstore.Top of Form