Student name: Ali M Rashid Student No: 180057319

Website Link: https://180057319.cs2410-web01pvm.aston.ac.uk/larabookstore/public/

Admin username: astonadmin@aston.com pass: astonadmin

User username: astonuser@aston.com pass: astonuser

Structure of system:

I used the Laravel framework to route my views with functions in the item and request controllers, I used the command line and Laravel functions like artisan to create the model and all the tables. I have used bootsrap to style the website, it is clean and simple to use. Users that arrive on the front screen where they can click a link to see the list of books on the database, if a user isn't registered they will see a basic list of books with details option, if a registered user clicks that link it will be sent to the list of books where they can add a book to the cart. The website uses gates to direct to users to the relevant pages.

Required functions implemented:

Public…

- can register as customer
- can see book list with title, category, price.
- Can see details of book with extra information like book cover image , stock and publish year

Registered Users…

- Can Log in or out
- can view list of books with basic details
- can view details of specific book by clicking on details button on the items list.
- Can check the cart
- Can add books to cart through book list
- Can delete a book from cart
- Can update the quantity of a book in cart
- Can place an order for a book

Staff can…

- Log in or out
- Can increase stock quantity of a book
- Can View order and their details
- Can Complete orders

Other functional features:

- Sensible names for URL
- Simple and neat webpages
- Adequate error reporting
- Pages have links to related pages and home page

see:
larabookstore/resources/views/layouts/app.blade.php line: 42-49
larabookstore /resources/views/books/show.blade.php line: 24-27

Security features

- Authentication is used to divert users to their relevant views, so a public user cannot access staff views
  See:
  larabookstore /app/Http/Controllers/BookController.php lines: 21-28, 39-43 …
  larabookstore /app/Http/Controllers/OrderController.php lines: 28-34

- Passwords are Hashed on the database

- All inputted requests have been validated and some fields guarded from mass injection
  See:
  larabookstore /app/book.php line: 17-20
  larabookstore /app/Http/Controllers/BookController.php lines: 56-65
  larabookstore /app/Http/Controllers/OrderController.php lines: 59-63

- Restricted file upload to only images with restricted size
  See:
  larabookstore /app/Http/Controllers/BookController.php lines: 64

- since Laravel is used all forms must have CSRF token to work
  see:
  larabookstore /resources/views/items/indexCustomer.blade.php line: 46

Stretchers Used

- Clever stock system, all input is restricted to prevent inserting a value above stock level
  When a book is added to cart while same book already exists in cart, the quantity is increased, and a repeated listing is not added.
  In the cart a user can override the stock level.
- A Mix of query builder and eloquent model methods are used to carry out CRUD functions. Query builder was used since loading times are more than 50% faster then eloquent methods
  See:

- Controller methods use gates to authenticate the user and direct them to the correct view, this prevents normal users from accessing admin pages by inputting the URL of the page.
  See: (all controller methods that direct to a view use gates)
  larabookstore /app/Http/Controllers/BookController.php lines: 21-28, 39-43 …
  larabookstore /app/Http/Controllers/OrderController.php lines: 28-34

Database structure

Disclaimer
I understand that a pivot table called (book_user) should have been used to store items in the cart instead of the orders table(see below), however I learnt this later, and due to shortness of time I continued to use the orders table to store cart items. To distinguish between orders and cart items I implemented a status field, when it is 0 that row is an item of the cart when it is 1 then that row is considered an order.

order table (many to many relationship):
book_id is foreign key references id in the book table
customer_id is foreign key references the id in user id