Description

It is an online book store or a book house where guest users can find and even buy a lot of great books with cheaper prices.

How it works

- When the customer visits the home page he will be displayed a collection of books.
- The customer will be able to browse and filter the books
- They will be able to add the books that they want to their cart and a cart icon will display how many items they have in the cart.
- The customer will be able to checkout and pay his order if he logged in and if not he will be redirected to the login page and after completing the login he will be redirected back to the checkout page.
- The customer will be able to modify the quantities of the items in the cart before the checkout process.
- If the customer completed the payment process successfully he will be redirected to a payment success page.

Targeted audience

This document will be created and used by the development team, project manager and targets people who love reading and students.

Overview

Account creation for Guests

- First Name
- Last Name
- Email
- Password

Portal screens

- The landing screen/product listing screen
    - Welcoming landing page
    - Books listing section
    - Filter Books Section
    - Search Books
    - Why us section

- Book Details Screen
    - The Book Details section
    - Similar Books Section

- Cart Screen
    - Cart Items listing
    - Pay & Checkout Button
    - Display total price

- Profile Screen
    - User profile picture
    - User informations section
    - Edit user information section
    - Listing user orders
- Payment Screen
    - Payment success message
- Order Details page
    - Order total price
    - List of items and quantities in the orders

- Sign in Screen
    - User login with email and password
    - Forgot password section
- Sign up Screen
    - User Signup with first name, last name, email and password.

- Categories Screen
    - List all categories
- Admin Pages
    - Books page
    - Orders page
    - Categories page

Requirements

Requirement	User Story	Importance	Notes
Listing Page/Landing Page	Users can check out a list of books. Users can add books to cart.	Must have	
Book Details Page	more information about the book. Users can add/remove the book to the cart.	Must have	
Shopping Cart	more information about Shopping cart	Must have	
Categories page	Displays all categories.	Should have	
Books page	Displays all books	Should have	
Sign up / Sign in pages	Sign up / sign in functionalities.	Must have	
User Profile page	Users can view and modify their information. View their orders.	Must have	
Admin pages	Admin can manipulate books, categories, orders, users	Must have	

Design guidelines

Roles

- Project manager:
  Lijeesh Majeed
- Team lead:
  Vimson Varghese
- Development team:
  Mohamed Farag, Ahmed Saladin, Mohamed Hussein, Mina Merzeq

Dependencies

- Dynamo db
- React
- Node
- Express
- Rest apis
- Aws sdk
- Axios
- UseQuery
- Websockets.io
- Paypal

Design Considerations

Operational environment

- App will work on browsers on pc, android and ios

Development methods

- Pair programing
- Extreme programing
- Test-driven development
- Agile: scrumban

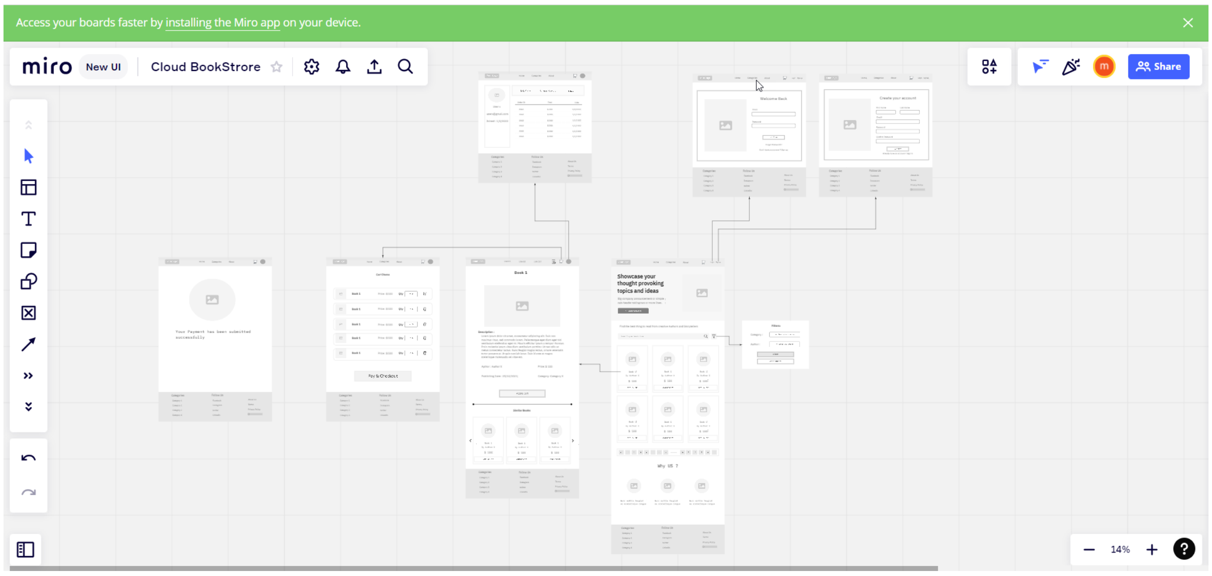
Architecture strategies

- Microservices

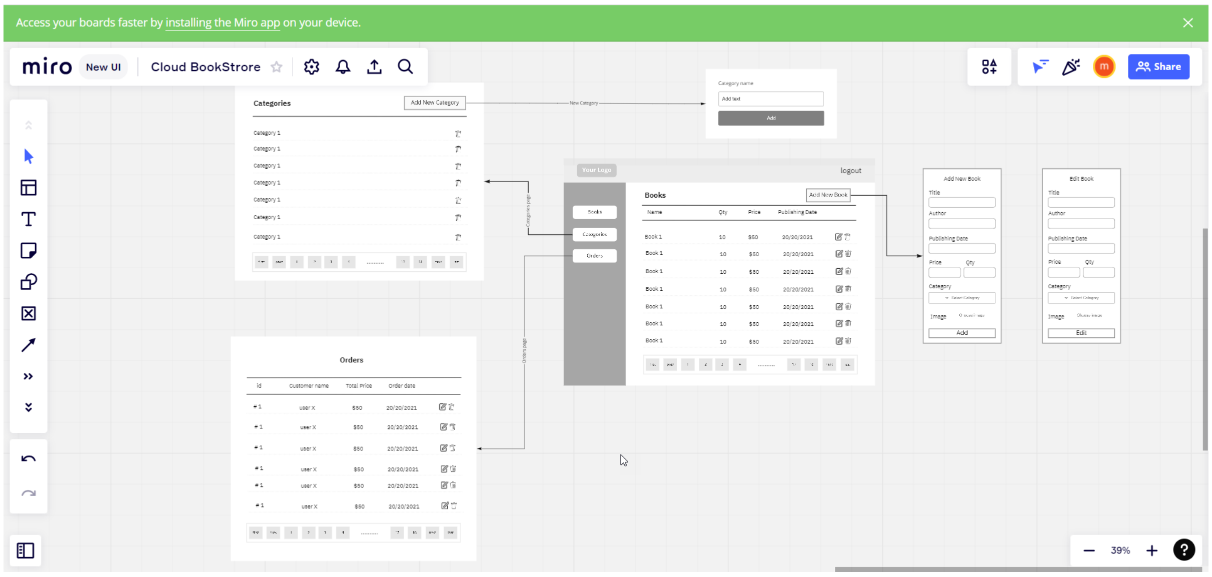
Frontend Architecture

Architecture link: https://miro.com/app/board/uXjVOQs7tKM=/

Customer pages

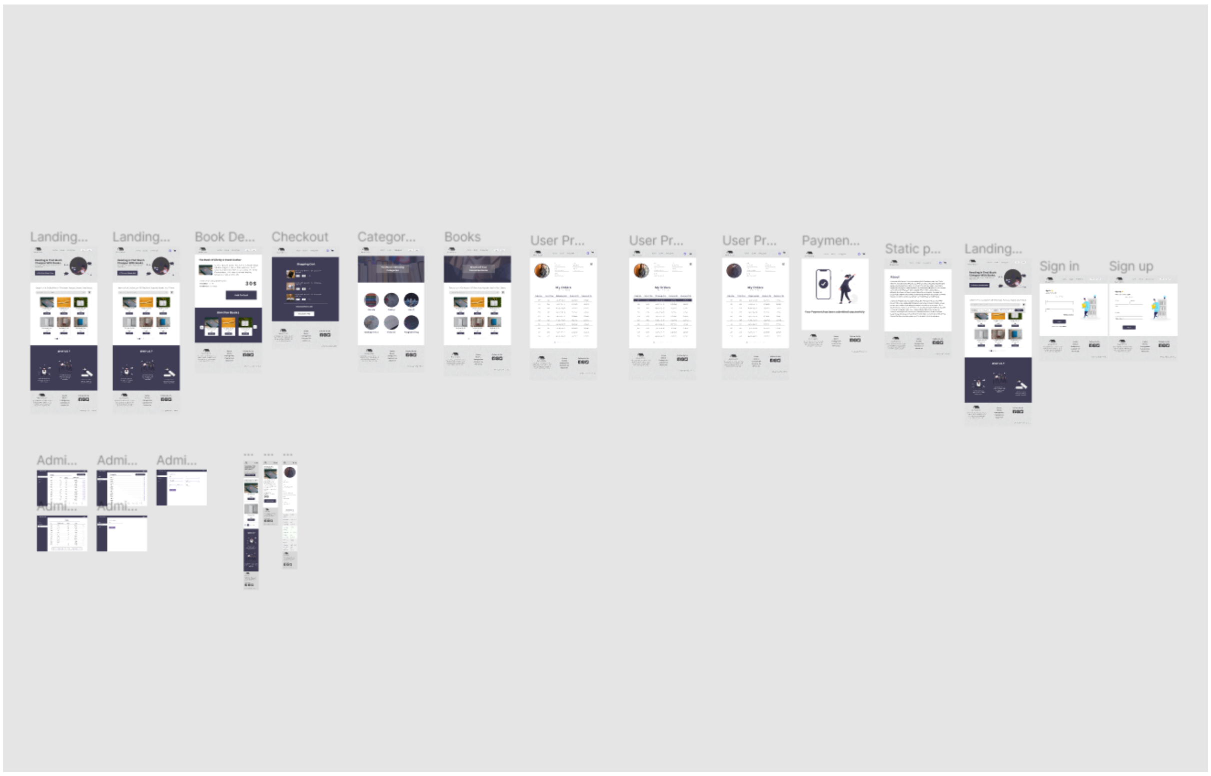

Access your boards faster by installing the Miro app on your device.
miro
New UI
Cloud BookStrore
Share
Your Payment has been submitted successfully
Pay & Checkout
Book 1
Showcase your thought provoking topics and ideas
Why US ?
Welcome Back
Create your account
14%

Admin pages

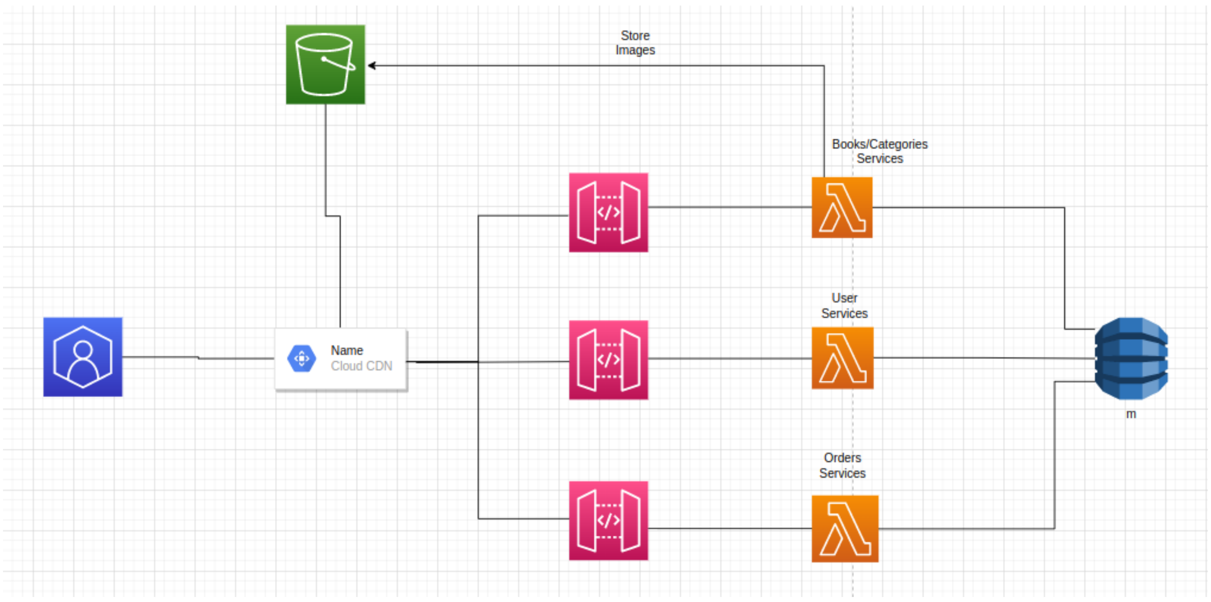


UI Design

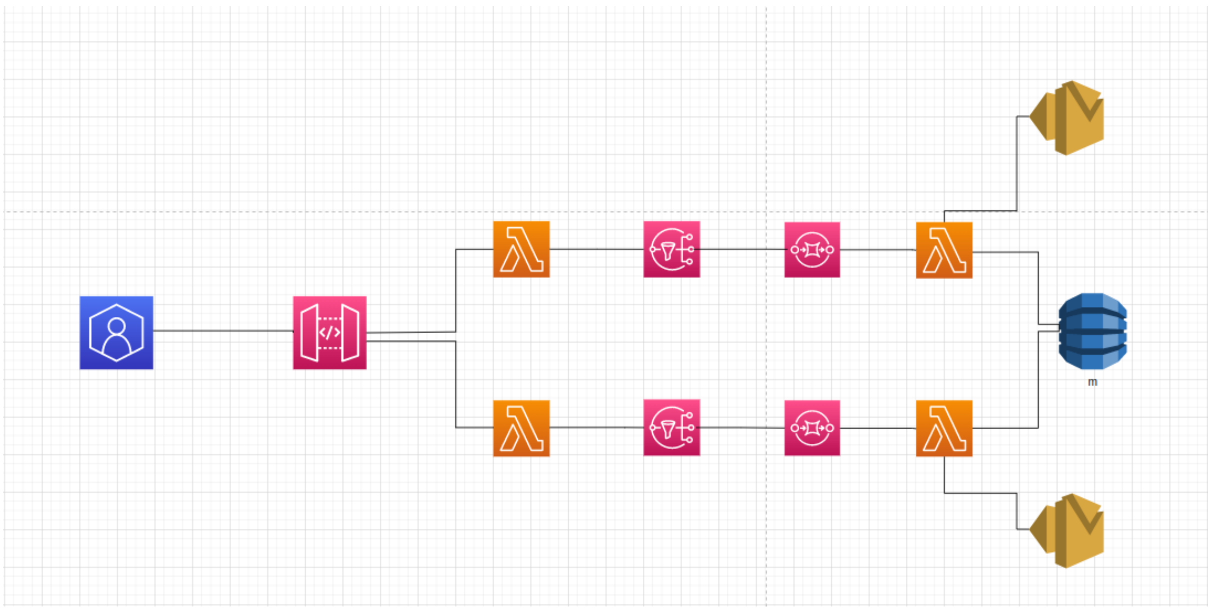
https://www.figma.com/file/zuGJTiRXhqC4MOeMLPwWeW/Book-house?node-id=0%3A1



Backend Architecture



Order Event-driven design



For order cancellation

- The request is sent to a lambda function with the id of the order and a message to the customer then the lambda function delete the order and publish a message to an sns topic and the sns topic has an sqs queue subscribes to it and pull messages whenever it is available and then trigger another lambda function that sends an email to the customer using ses

For making new order

- The request is sent to a lambda function with the order data and the lambda function creates the order in the database with pending status and calls the PayPal api to get a payment id.
  Then the lambda function is triggered again once the payment is done and it publish a message to an sns topic that has an sqs queue subscribed to it and trigger another lambda function that will evaluate the order and then send an email to the customer using ses to inform him whether the order has been confirmed or rejected

Database Models

Entity: Books

~ Get all the books (PK = books and SK begins with)

~ Get all books belongs to a category (LSI-2-PK ==> book#category and begins with book#)

~ Get a single book (PK = books and SK = book#author#bookname)

~ Get all the books in the DB started with character 'o' (LSI-1-PK == books & LSI-1-SK begins with book#C)

Attributes:

slug

title

author

category

description

image

price

quantity

publishing_date

created_at

Updated_at

get(where pk = books & sk begins with book#)

PK ==> books

SK ==> book#slug

LSI-1-PK ==> books

LSI-1-SK ==> book#category#slug

LSI-2-PK ==> books

LSI-2-SK ==> book#title

Get where pk = books and sk begins with book#

Get where pk = books and sk = book#atomic-habits

Get where LSI-1-PK = books and LSI-1-SK begins with book#development

Get where LSI-2-PK = books and LSI-2-SK begins with book#b

—---

Entity orders

- Structure
    - order_id
    - User_id
    - Address //paypal
        - Name
        - Building_number
        - Street
        - City
        - Country
        - postal_code
    - Status
    - createdAt
    - Books
        - book_id
        - Qty
        - price

- Access pattern
    - Get all the orders
    - Get the orders for specific user
    - Get order details

- pk   ====> order
- sk   ====> order#<order_id>
- LSI-1-PK ==> orders
- LSI-1-SK ==> order#<user_id>

Get where pk=order & sk begins with order#

Get where pk= order & sk = order#565666

Get where LSI-1-PK = order & LSI-1-SK = order#6566333

—---

## Users

## Access patterns

- Get Single User Entity: Users

Attributes:

uder_id

first_name

last_name

email

password

phone

created

PK => users

SK => user#user_id

Get where pk = users & sk = user#656336

—---

### Category:

# Access patterns:

~ Get all category (PK = category)

~ Get single category by title (PK = category and SK= category#title)

# Entity: Category

Attributes:

slug

title

photo

PK ==> category

SK ==> category#slug

—---

### Cart:

```
# Access patterns:

~ Get cart items for user (PK = userId and SK begins with
cart#)


# Entity: Cart

Attributes:

     {

          book_id

          quantity

     }


PK ==> userId

SK ==> cart#datetime#author#bookname
```

——--

Pk

Sk

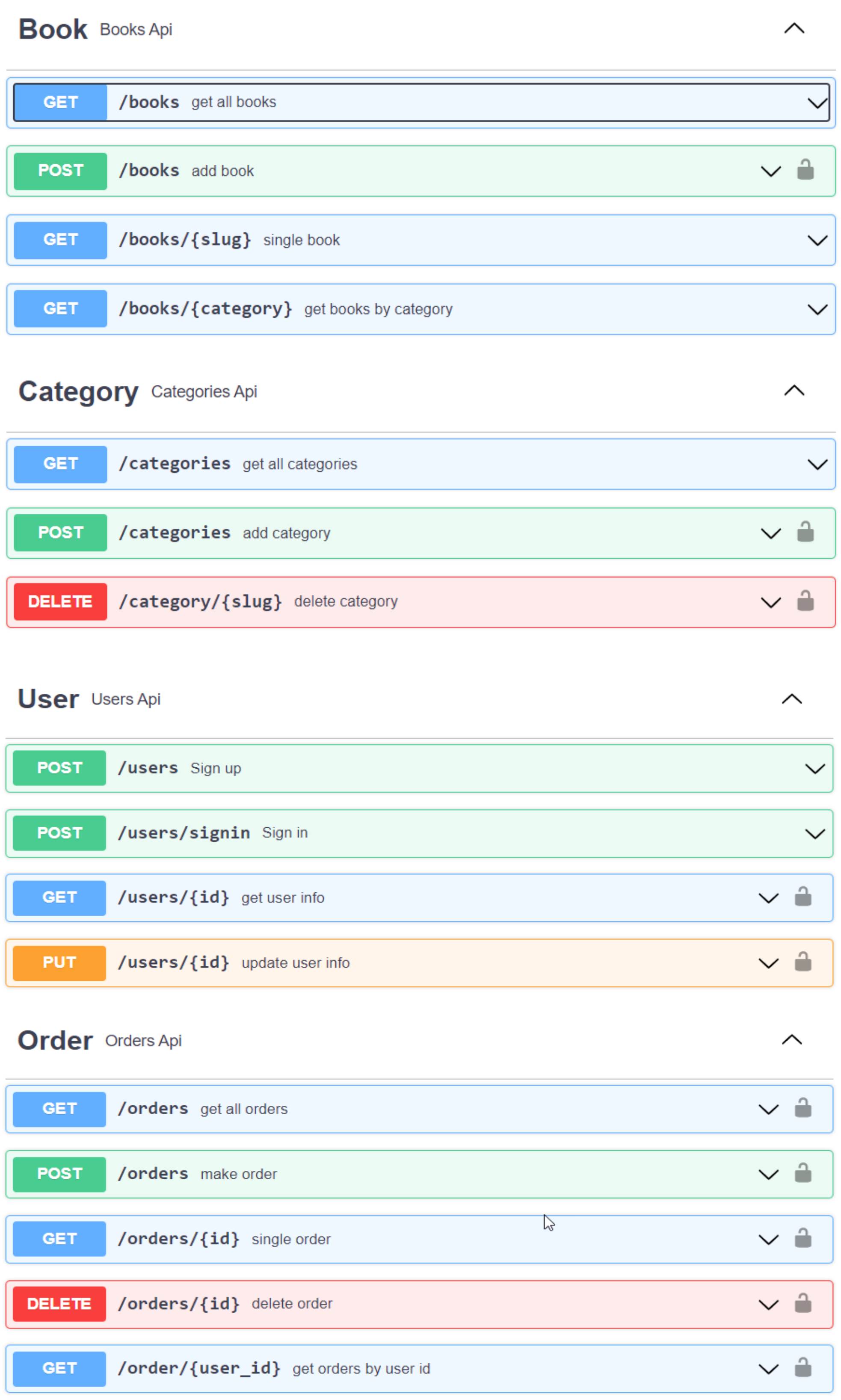
Lsi1pk

Lsi1sk

lsi1-index

——--

API Endpoints


Book Books Api

GET `/books` get all books

POST `/books` add book

GET `/books/{slug}` single book

GET `/books/{category}` get books by category

Category Categories Api

GET `/categories` get all categories

POST `/categories` add category

DELETE `/category/{slug}` delete category

User Users Api

POST `/users` Sign up

POST `/users/signin` Sign in

GET `/users/{id}` get user info

PUT `/users/{id}` update user info

Order Orders Api

GET `/orders` get all orders

POST `/orders` make order

GET `/orders/{id}` single order

DELETE `/orders/{id}` delete order

GET `/order/{user_id}` get orders by user id

Use Cases

User

- Browsing the books
    - Overview:

      This scenario describes a user is browsing all books
    - Preconditions:

      From the home page the user can press the “discover books now” button or even scroll down to find a group of books and navigate from group to group using the pagination buttons.
      User also can do that by clicking on the books nav tab in the nav bar and follow the same things

- Searching for specific book by name, author and category
    - Overview:

      This scenario describes a user is looking for specific book
    - Preconditions:

      From the home page the user can scroll down to find a big search box and a filter button beside it.

      The user can start typing inside the search box and the results will appear in front of him as for matching book names

      The user can press the filter button and choose to filter with author or category
      For the category he can also press the category nav tab to move to brand new page holding all categories and pressing on any category will move again for a new page holding all books under that category

- Get book details and add to cart and remove from it
    - Overview:

      This scenario describes a user is getting book details and adding book to cart and remove from
    - Preconditions:

      While user is browsing books he can press on a book’s “read more” button to move to book details page and form there he can add or remove book from cart using “add / remove” button
    - Notes:

      When adding or removing there is an indicator in the nav bar on the right for how many items are there in the cart

      User can add to cart even without registering or signing in

- Registering and signing in
    - Overview:

      This scenario describes a user is registering or signing in
    - Preconditions:

      From any page user can press the register or sign in buttons in the nav bar to complete the process

- Viewing profile info and orders details and edit info
    - Overview:

      This scenario describes a user is viewing and editing profile info and viewing orders details

- Preconditions:
  From any page user can press on his icon in the nav bar to move to profile page to view info press edit tab to show edit form
  Also scrolling down he will find his orders and details

- Completing purchase
  - Overview:
    This scenario describes a user is completing order and payment
  - Preconditions:
    From any page user can press on the cart icon in the nav bar to move to cart page to view items to order
    User can modify quantity or remove items from cart then fill the address form and press “paypal” button and complete the process
  - Notes:
    User should login first before being able to complete the payment process

Admin

- Registering and signing in
  - Overview:
    This scenario describes an admin is registering or signing in
  - Preconditions:
    Same as user
  - Notes:
    When admin login he is move to a brand new page or dashboard specified for him

- Manipulating books
  - Overview:
    This scenario describes an admin is viewing, adding, editing and deleting books
  - Preconditions:
    From dashboard admin can press books tab from the left nav bar to navigate to books page
    From the top right “add new book” button he can add new book and from the the books table he can view books and edit or delete a single book

- Manipulating categories
  - Overview:
    This scenario describes an admin is viewing, adding and deleting categories
  - Preconditions:
    From dashboard admin can press categories tab from the left nav bar to navigate to categories page
    From the top right “add new category” button he can add new category and from the the categories table he can view and delete categories

- Managing orders
    - Overview:

      This scenario describes an admin is viewing and maybe cancelling incoming orders
    - Preconditions:

      From dashboard admin can press orders tab from the left nav bar to navigate to orders page

      From the the orders table he can view and cancel orders