

# FINAL PROJECT REPORT

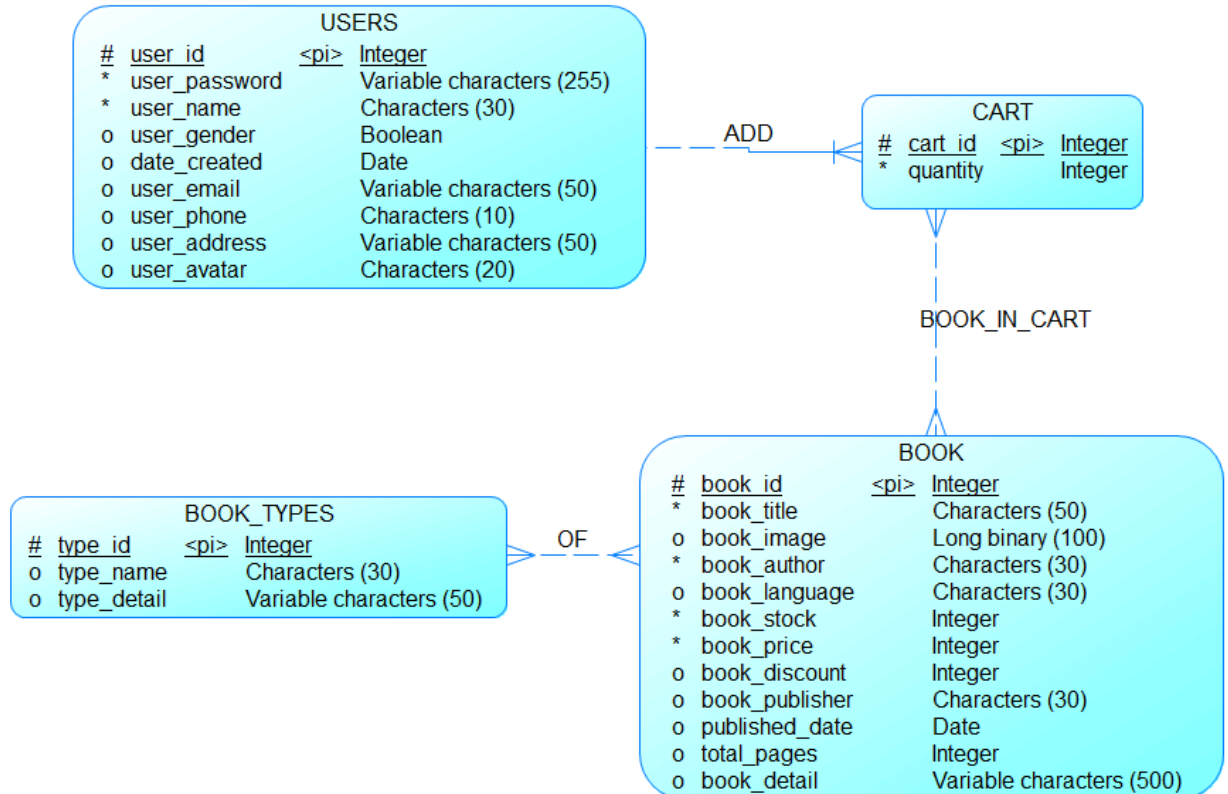
SEMESTER 1, ACADEMIC YEAR: 2024-2025

CT313H: WEB TECHNOLOGIES AND SERVICES

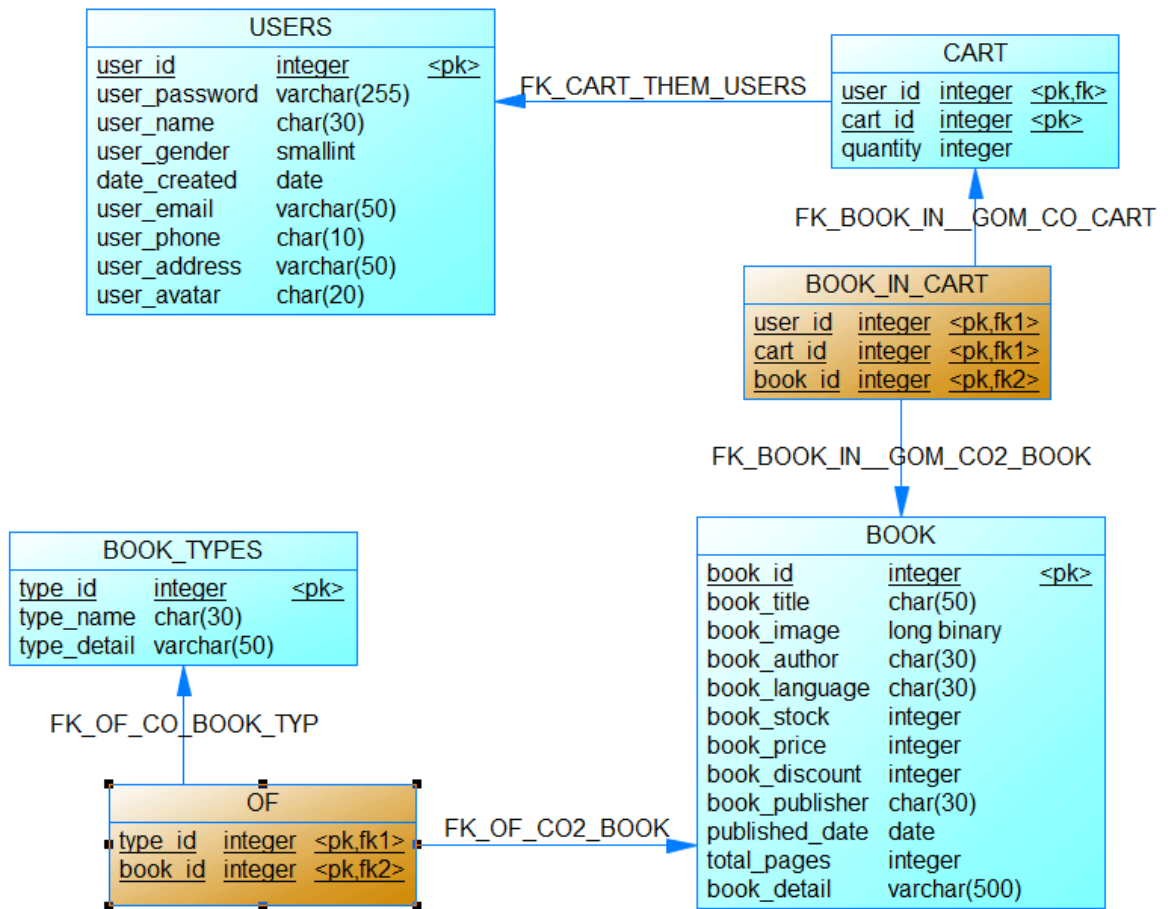
- **Project/Application name:** BOOKSHOP WEBSITE
- **GitHub links:** <https://github.com/24-25Sem1-Courses/ct313h01-project-nhan295>
- **Youtube link demo:** <https://www.youtube.com/watch?v=Ld0grySpQ2w>
- **Student ID 1:** B2105663
- **Student Name 1:** Ton Thi Ngoc Chau
- **Student ID 2:** B2111940
- **Student Name 2:** Tran Thi Hong Nhan
- **Class/Group Number:** CT313HM01

## I. Introduction

- A comprehensive platform for users to browse, search, and purchase books online. It features well-organized categories, personalized recommendations based on user preferences, and a seamless, secure shopping experience designed to make book discovery and purchasing enjoyable and hassle-free.
- CDM:



- PDM:



## II. Details of implemented features

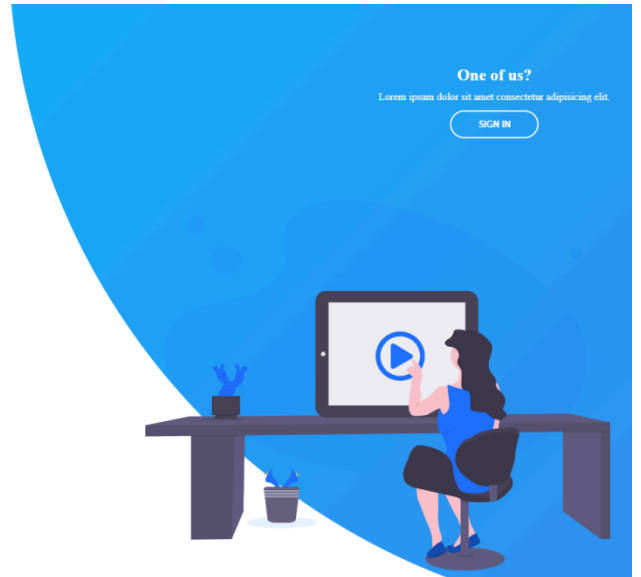
### 1. Sign up feature:

- **Description:** The sign up feature allows users to create a new account on the website. Users are required to provide personal information such as a username, email, password, etc. Once the sign up process is completed successfully, their account details are stored in the database, and redirect to the homepage.
- **Screenshots:**

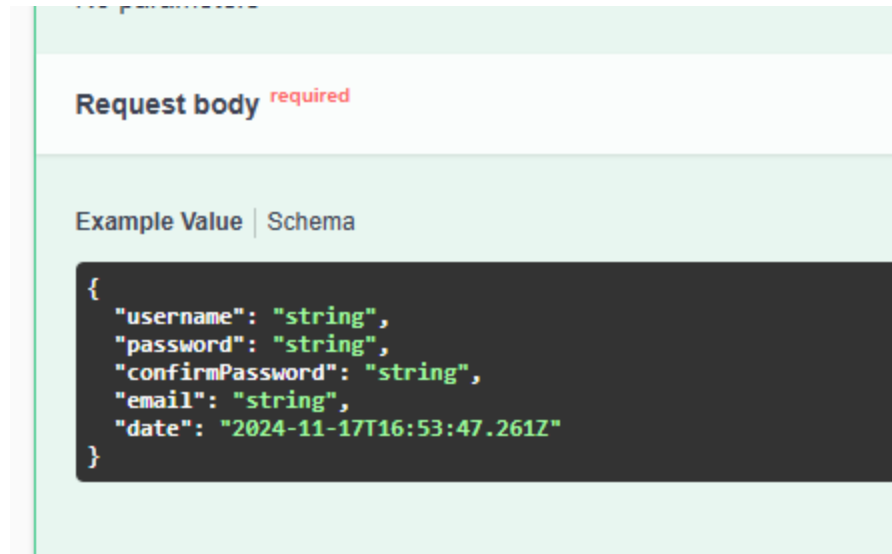
**Sign up**

**SIGN UP**

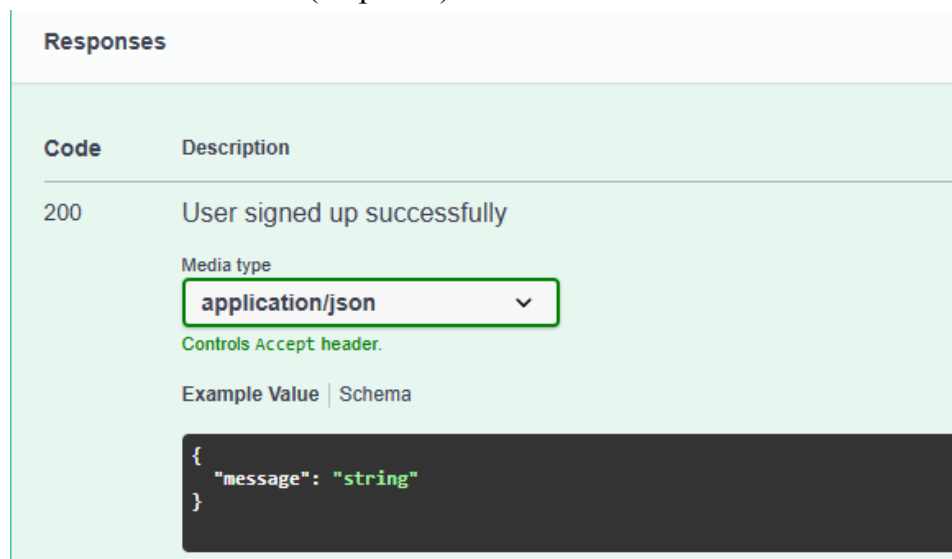
Or Sign up with social platforms



- **Implementation details:**
  - + **Backend library:**
    - bcrypt
    - express-session
    - knex
    - dotenv
    - crypto
  - + **Frontend library:**
    - Vue
    - Vue-Router
    - Font-Awesome
    - Google Fonts
  - + **Sign up API:**
    - Endpoint: /api/v1/users/signup
    - Method: POST
    - Data format sent (request body):



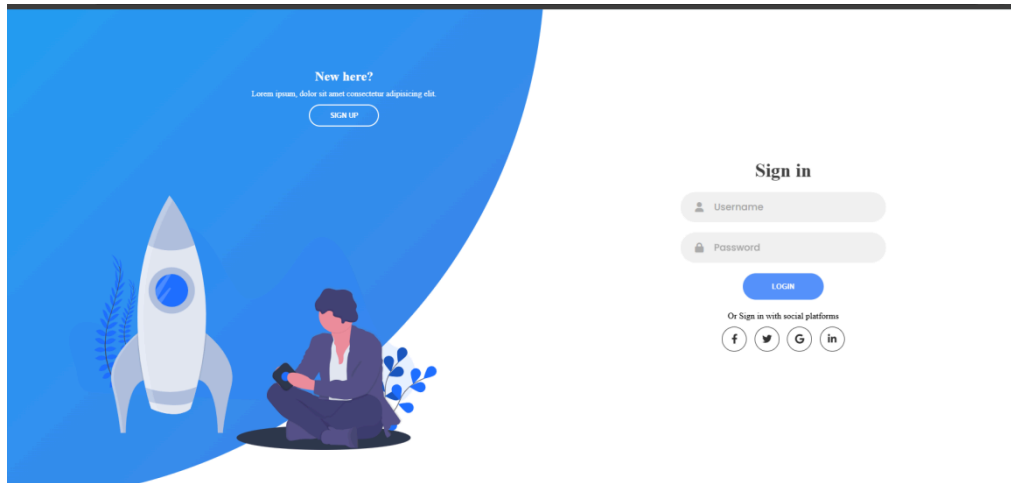
- Data format received (response):



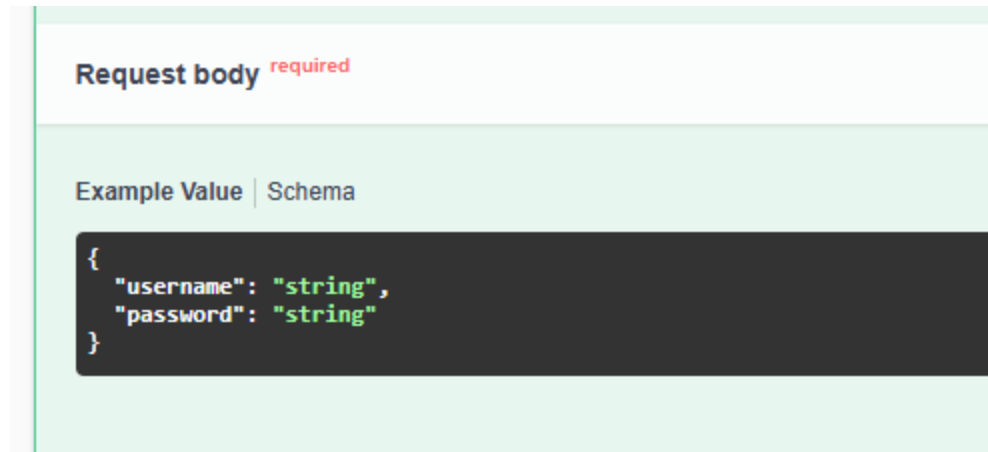
- + **This feature stores data:** When a user signs up, their information is stored in the **users** table in the database
- + **Client-side**
  - signUpData
  - signUpMessage
  - isActive
  - currentDateTime
  - useRouter

## 2. Sign in feature:

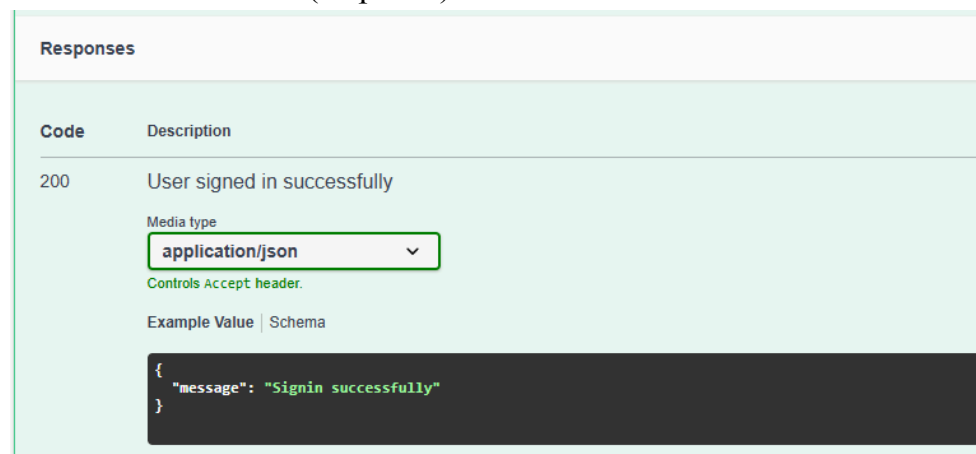
- **Description:** The sign-in feature allows users to log into their existing accounts on the website. Users need to provide their username and password. Upon successful authentication, they gain access to their account and can be redirected to the homepage.
- **Screenshots:**



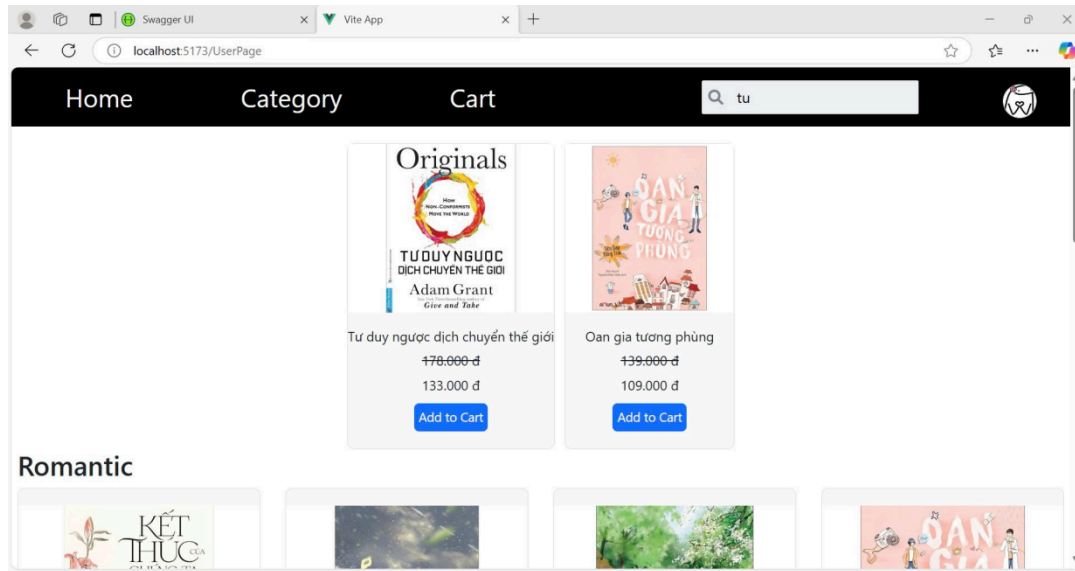
- **Implementation details:**
  - + **Backend library:**
    - bcrypt
    - express-session
    - knex
    - dotenv
    - crypto
  - + **Frontend library:**
    - Vue
    - Vue-Router
    - Font-Awesome
    - Google Fonts
  - + **Sign in API:**
    - Endpoint: /api/v1/users/signin
    - Method: POST
    - Data format sent (request body):



- Data format received (response):



- + **This feature read data:** The sign in operation checks the provided credentials against the existing user data in the **users** table.
  - + **Client-side**
    - signUpData
    - signUpMessage
    - isActive
    - useRouter
- 3. Search book feature:**
- **Description:** The Search Flashcard feature allows users to quickly find specific book based on Book title. This feature provides a convenient search bar where users can type in search terms, and the application will instantly filter and display relevant books from all Book. Results appear in real-time as users type.
  - **Screenshots:**



- **Implementation details:**

+ **Backend library:**

- Express
- Knex

+ **Frontend library:**

- Vue
- Axios
- Emits
- Vue-router

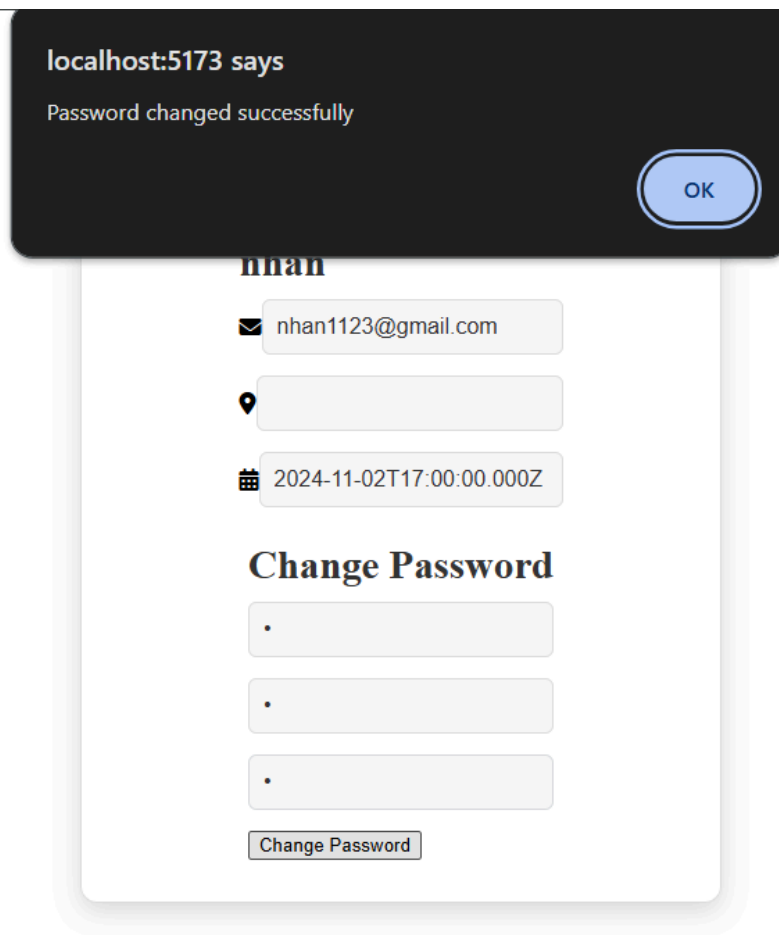
+ **Search Book API:**

- Endpoint: `/api/v1/book/search/{searchText}`
- Method: GET
- Data format sent (request body): No specific data is sent in the body for search book
- Data format received (response):

Example Value | Schema

```
[
  {
    "id": 0,
    "title": "string",
    "author": "string"
  }
]
```

- + **This feature doesn't read/store data**
- + **Client-side**
  - searchInput
  - searchQuery
- 4. **Update user password**
  - **Description:** When a user logs in, they can view the account information of the recently logged-in account, along with an option to change the password. To change the password, the user is required to enter the current password and the new password.  
After successfully changing the password, the system will display a notification.
  - **Screenshots:**



- **Implementation details:**
  - + **Backend library:**
    - Express
    - Knex
    - bcrypt



+ **Frontend library:**

- Vue
- Vue-router

+ **User data API:**

- Endpoint: /api/v1/book/changePassword
- Method: PUT
- Data format sent (request body):

Request body **required**

Example Value | Schema

```
{
  "currentPassword": "string",
  "newPassword": "string"
}
```

- Data format received (response):

Responses

Code	Description
200	Password changed successfully
400	Invalid current password
500	Error changing password

Media type  
application/json

Controls Accept header.

Example Value | Schema

```
{
  "message": "Password changed successfully"
}
```

+ **This feature store data** and update `password` column in the progress table

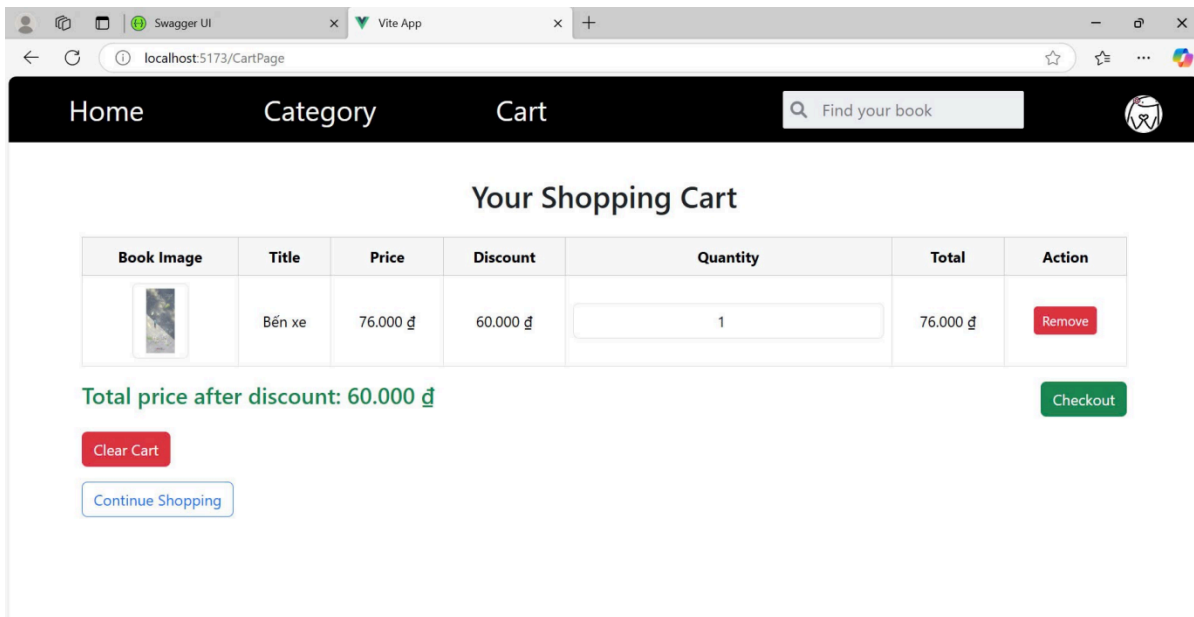
+ **Client-side**

- isChangingPassword
- currentPassword

- newPassword
- confirmPassword
- successMessage
- error message

## 5. Add Book to Cart feature

- **Description:** Users can add items to the cart from the product listing or product details pages.
- **Screenshots:**



- **Implementation details:**
  - + **Backend library:**
    - express
    - knex
  - + **Frontend library:**
    - Vue
    - Vue-router
  - + **Add Book API:**
    - Endpoint: /api/v1/cart/add
    - Method: POST
    - Data format sent (request body):

Example Value	Schema
<pre>{   "user_id": 0,   "book_id": 0,   "quantity": 0 }</pre>	

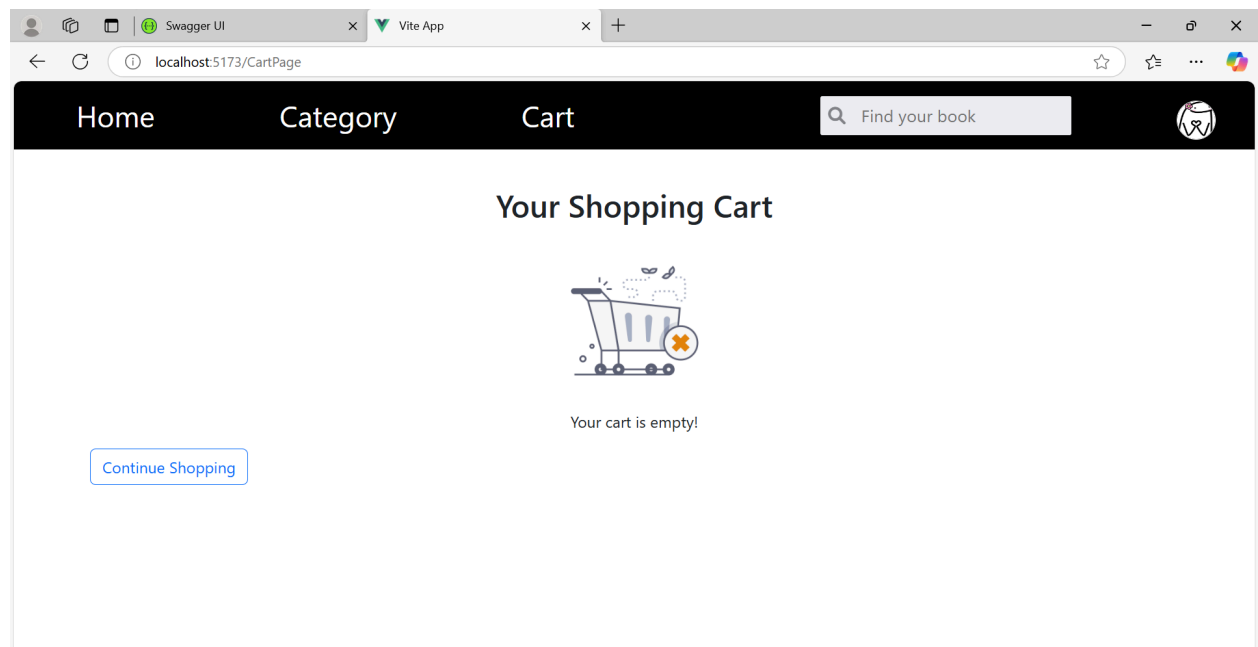
- Data format received (response):

Responses	
Code	Description
201	Item added to cart successfully

- + The data is store in a CARTS table:

## 6. Delete Book from Cart feature

- **Description:** Provides the option to remove individual items from the cart and display an empty cart with a Home navigation button.
- **Screenshots:**



- **Implementation details:**
  - + **Backend library:**
    - express

- knex

+ **Frontend library:**

- Vue
- Vue-router

+ **Add Book API:**

- Endpoint: /api/v1/cart/delete
- Method: DELETE
- Data format sent (request body):

Example Value | Schema

```
{
  "user_id": 0,
  "book_id": 0
}
```

- Data format received (response):

Responses	
Code	Description
200	Cart item deleted successfully