



Phan Nhat Linh GCD2016 35 GCD0905 ASM2 1670

Working in groups (Trường Đại học FPT)



Scan to open on Studocu

ASSIGNMENT 2 FRONT SHEET

Qualification	BTEC Level 5 HND Diploma in Business		
Unit number and title	Unit 30: Application Development		
Submission date		Date Received 1st submission	
Re-submission Date		Date Received 2nd submission	
Student Name	Phan Nhat Linh	Student ID	GCD201635
Class	GCD0905	Assessor name	Nguyen Van Loi
Student declaration I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.			
		Student's signature	Linh

Grading grid

P4	P5	P6	M3	M4	M5	D2	D3

☐ **Summative Feedback:**☐ **Resubmission Feedback:****Grade:****Assessor Signature:****Date:****Internal Verifier's Comments:****Signature & Date:**



ASSIGNMENT 2

Subject: Application Development (1670)

Table of Contents

I/ INTRODUCTION	7
II/ PEER REVIEW AND FEEDBACK ANALYSIS (P4).....	7
1. Formal questionnaire to review the business application, problem definition statement, proposed solution, and development strategy	7
2. Collect reviews feedback.....	9
3. Interpret peer-review feedback	14
4. Evaluate any new insights, ideas, or potential improvements	15
4.1 Extensive book selection	15
4.2 Offering book recommendations	16
4.3 Make the website to be responsive	17
4.4 Send a notification when customers order	18
4.5 Show confirm notification before removing	19
III / APPLICATION DEVELOPMENT (P5).....	21
1. Folder structure of the application	21
2. Source code samples of the application with explanation	24
2.1 Model	24
2.2 View	27
2.3 Controller.....	30
3. Final screenshots of the application.....	33
4. Screenshots of using GitHub to manage the source code	41
5. Screenshots of using IIS or Azure for the application deployment.....	41
IV/ APPLICATION EVALUATION (P6).....	44
1. Review the performance of the application	44
1.1 Test case	44
1.2 Test result	46
2. Conclude whether the application adapts all requirements or it needs to be improved later	57
3. Analyze the factors that influence the performance of the application.....	58
4. Evaluate the strengths and weaknesses of the application	59

Table of Figures

Figure 1: Survey questions (1)	8
Figure 2: Survey questions (2)	8
Figure 3: Survey questions (3)	9
Figure 4: Feedback from Question 1	9
Figure 5: Feedback from Question 2	10
Figure 6: Feedback from Question 3	10
Figure 7: Feedback from Question 4	11
Figure 8: Feedback from Question 5	11
Figure 9: Feedback from Question 6	12
Figure 10: Feedback from Question 7	12
Figure 11: Feedback from Question 8	13
Figure 12: Feedback from Question 9	13
Figure 13: Book selection	16
Figure 14: Offering book recommendations	17
Figure 15: Responsive website	18
Figure 16: Sending notification	19
Figure 17: Show confirmation	20
Figure 18: Folder structure of the application	21
Figure 19: Shop role structure of the application	23
Figure 20: Book class	24
Figure 21: Category class	25
Figure 22: Order class	26
Figure 23: Category (index)	27
Figure 24: Home (index)	28
Figure 25: Order (index)	28
Figure 26: Product (index)	29
Figure 27: Category controller	30
Figure 28: Home controller	30
Figure 29: Order controller	31
Figure 30: Product controller	32
Figure 31: Dashboard	33
Figure 32: List order and top product	34
Figure 33: Manage category	35
Figure 34: Request add new category	36

Figure 35: Manage book.....	37
Figure 36: Add a new book.....	38
Figure 37: Customer detail and order	39
Figure 38: Other order.....	40
Figure 39: Using GitHub to manage the source code	41
Figure 40: Create new web app	42
Figure 41: Select method publish.....	42
Figure 42: Select existing service.....	43
Figure 43: Result.....	43
Figure 44: Measure Performance.....	56

I/ INTRODUCTION

In assignment 1, my team members and I designed and analyzed the bookstore e-commerce website with 3 main roles as well as providing various diagrams, technologies, and mockups relating to the database, function, and interface of the system. The system is expected to support many modern and convenient functions for customers to be able to buy books rapidly. Regarding store owners and admin, it also offers a range of management tools relating to accounts, books, categories, and orders, making them easily control the system.

In assignment 2, a document with 3 sections was written to provide readers a piece of insightful information about the bookstore system. In order to explain the system, I will present it in the following order:

- Peer review and feedback analysis
- Application development
- Application evaluation

II/ PEER REVIEW AND FEEDBACK ANALYSIS (P4)


1. Formal questionnaire to review the business application, problem definition statement, proposed solution, and development strategy

Generally, my team's purpose is to develop an e-commerce application that can provide a convenient and enjoyable shopping experience for book lovers. However, my team is lacking knowledge about satisfying customers' experience and ways to improve their trust in the system. Therefore, a formal questionnaire will be designed to assist my team to gain new insights, ideas, and potential improvements to my system. Below is my formal questionnaire:

survey about the FPT bookstore e-commerce

This survey is only used to studies, please give us your honesty feedbacks

ndthien1112@gmail.com [Chuyển đổi tài khoản](#)

 Không được chia sẻ

1. On a scale of 1 to 5, are you satisfied with the system's functionality?

1 2 3 4 5

Strongly disagree ☐ ☐ ☐ ☐ ☐ Strongly agree

2. On a scale of 1 to 5, how would you rate the system's performance?

1 2 3 4 5

Very bad ☐ ☐ ☐ ☐ ☐ Very good

3. On a scale of 1 to 5, how do you like the interface of the system?

1 2 3 4 5

Strongly dislike ☐ ☐ ☐ ☐ ☐ Strongly like

Figure 1: Survey questions (1)

4. Which features are important to you when shopping for books online?

☐ User reviews and ratings

☐ Fast and reliable shipping

☐ Stunning interface

☐ Offering convenient functions

5. What factors influence your decision to purchase books online?

☐ Convenience

☐ Price

☐ Selection

☐ Availability

6. How do you typically search for books when shopping online?

☐ By title

☐ By author

☐ By genre

☐ By keyword

Figure 2: Survey questions (2)

7. Which of the following pain points have you experienced when shopping for our books online?

☐ Difficulty finding desired books

☐ Confusing checkout process

☐ Poor customer service

☐ Bad performance

☐ Everything keeps well

8. What features do you look for when shopping for books online?

Câu trả lời của bạn

9. Do you have any additional feedback or recommendations for our team as we continue to develop our bookstore e-commerce system?

Câu trả lời của bạn

Figure 3: Survey questions (3)

2. Collect reviews feedback

Result of the question number one is quite positive, with more than 80% of people satisfied with the functions of the system.

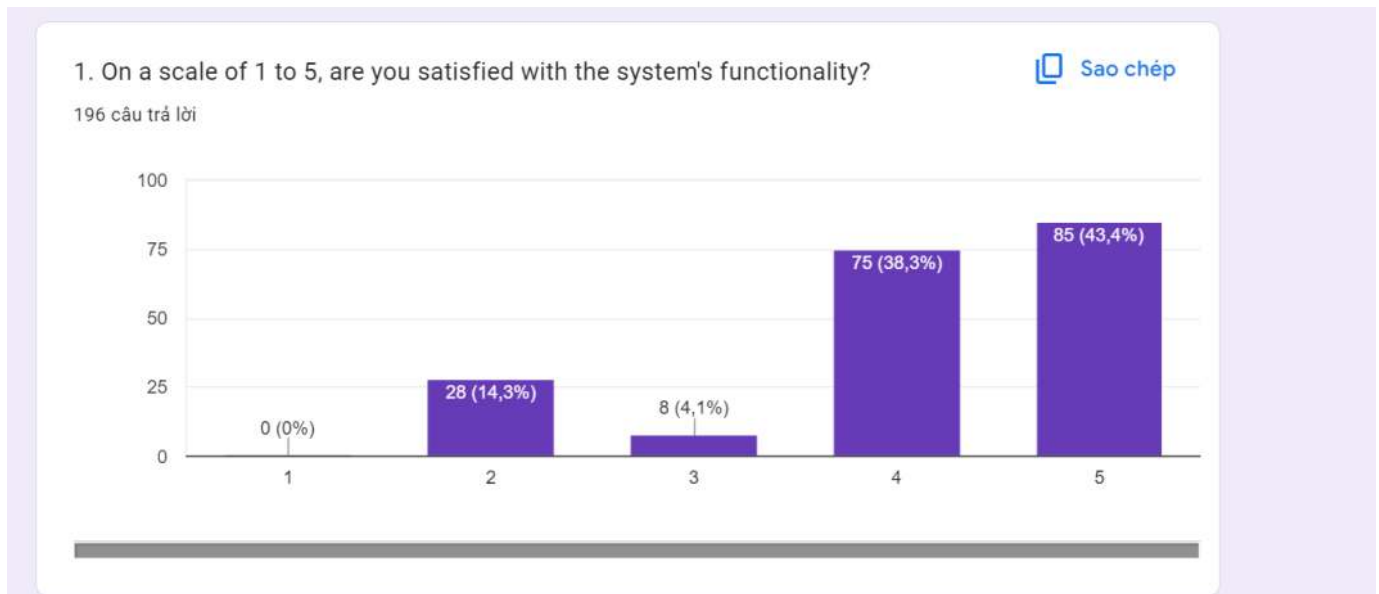


Figure 4: Feedback from Question 1

Result of the question number two is also positive, with more than 81% of participants rating 4 or 5 for the performance of the system.

2. On a scale of 1 to 5, how would you rate the system's performance?

Sao chép

196 câu trả lời

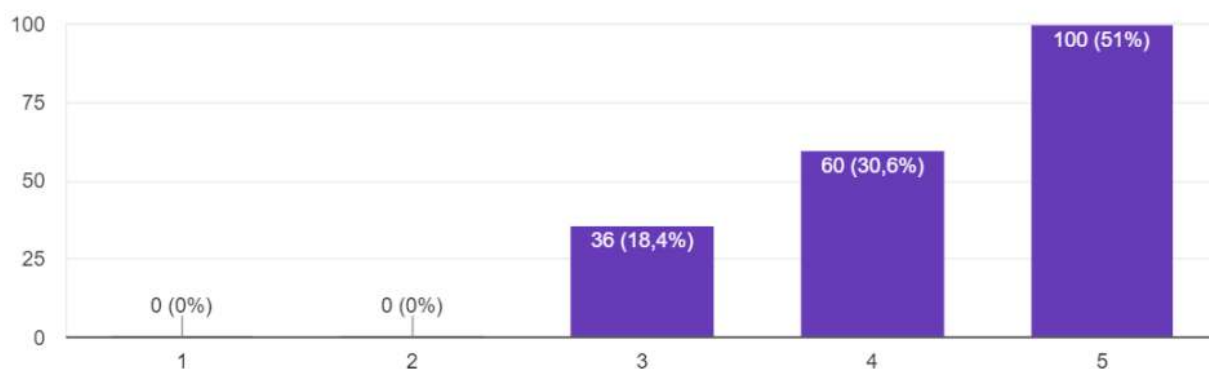


Figure 5: Feedback from Question 2

In question 3, we gain 89% of the participants who like the interface of the system

3. On a scale of 1 to 5, how do you like the interface of the system?

Sao chép

196 câu trả lời

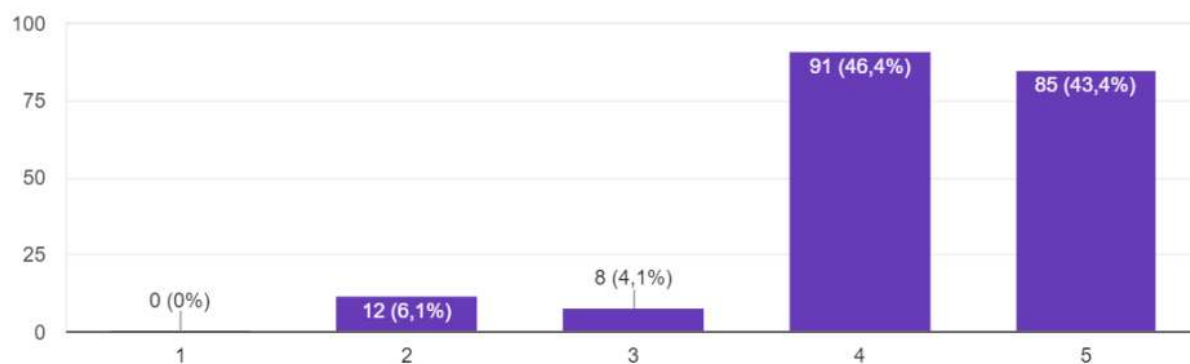


Figure 6: Feedback from Question 3

In question 4, we see that 66% of people want the system to offer convenient functions and 23,5% of people want to see the feedback of other customers before purchasing books

4. Which features are important to you when shopping for books online?

Sao chép

196 câu trả lời

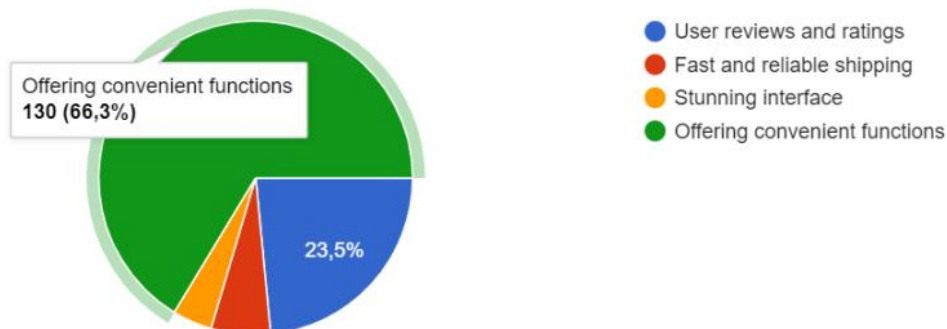


Figure 7: Feedback from Question 4

In question 5, we realize that convenience and price are two key factors affecting customers' intention to buy books

5. What factors influence your decision to purchase books online?

Sao chép

196 câu trả lời

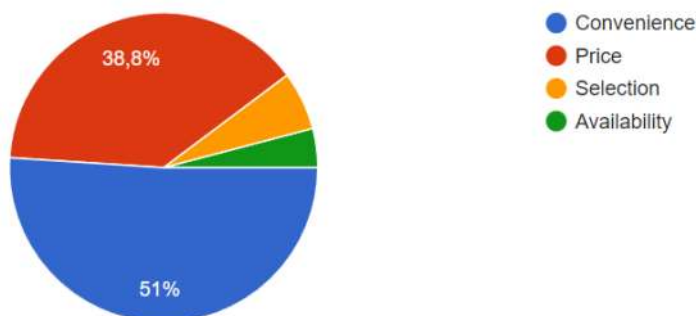


Figure 8: Feedback from Question 5

In question 6, most people like to search their desired books by title or by genre, with 53,6% for the title and 36,2% for the genre.

6. How do you typically search for books when shopping online?

Sao chép

196 câu trả lời

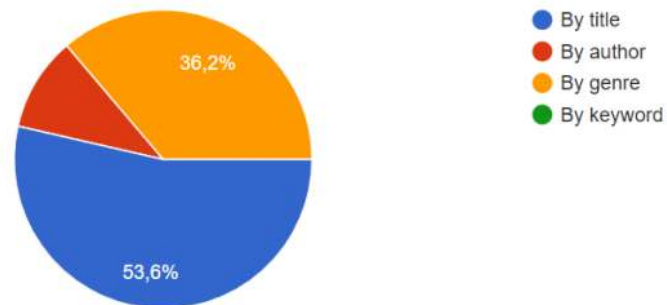


Figure 9: Feedback from Question 6

In question 7, in addition to the fact that most people are satisfied with the system, there is also some feedback about the bad performance of the system (19,4%) and some other responses.

7. Which of the following pain points have you experienced when shopping for our books online?

Sao
chép

196 câu trả lời

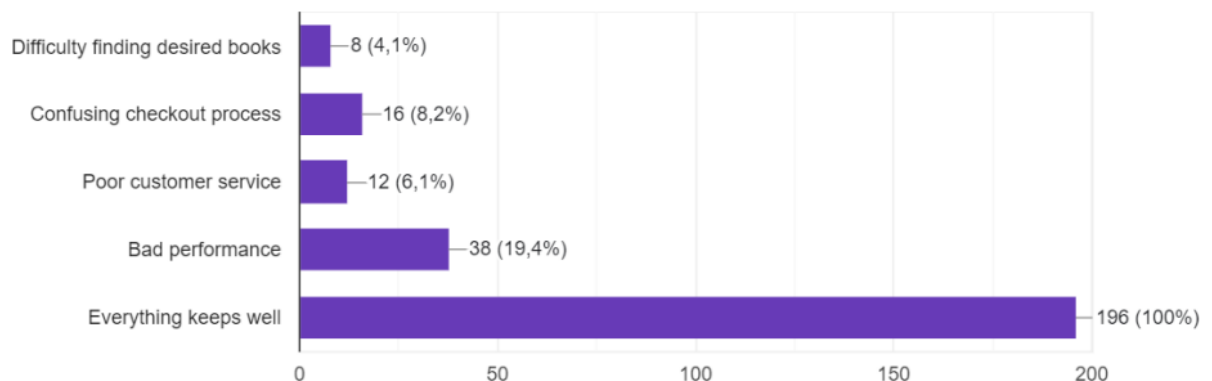


Figure 10: Feedback from Question 7

In question 8, we gain some features that people always look for when using the system:

8. What features do you look for when shopping for books online?

137 câu trả lời

Mobile-friendly design and user experience

User reviews and ratings

Competitive pricing and discounts

Personalized book recommendations

Ability to track orders and receive updates

Extensive book selection

Figure 11: Feedback from Question 8

In question 9, we gain many useful recommendations from participants:

9. Do you have any additional feedback or recommendations for our team as we continue to develop our bookstore e-commerce system?

137 câu trả lời

Send a notification when users successfully make an order to ensure that their orders were processed

Ensure that the website is mobile-friendly and has a responsive design that adjusts to different screen sizes.

Make sure the search functionality is robust and allows users to easily find the books they're looking for by title, author, genre, or keyword.

Provide a seamless and easy-to-use checkout process that includes multiple payment options

Offer fast and reliable shipping options with tracking and delivery notifications to keep users informed about the status of their orders.

Figure 12: Feedback from Question 9

3. Interpret peer-review feedback

Interpret question 1: Through the results obtained from the first question, with 38,3% of participants selecting 4 grade and 43,4% among them selecting max grade - grade 5, we can conclude that the functions of the system met customers' basic demands. However, there are also a few people who think the system is lacking some different functions, so we will provide more functions in the next update.

Interpret question 2: The bar graph shows positive feedback from the participants, with 51% of them choosing level 5 and 30.6% of them choosing level 4. We conclude that the performance of the system is quite optimal, so we will continue to keep the performance constant.

Interpret question 3: From the results obtained from the survey, we concluded that the interface of the system provided a very good user experience, with 46% of participants choosing a score of 4, and 43% of participants choosing a score of 5.

Interpret question 4: Through feedback from participants, we found that the convenient functions of the bookstore system are the most important feature that helps our e-commerce website attract customers. Therefore, we will provide more functions that make customer purchase books easily and quickly in the next update. Besides, user review is also an essential feature that customers want my system to offer. So, we will take a plan in the near future to develop this function.

Interpret question 5: In question 5, most people agree that the convenience of the system and the price of the books are the two factors that influence their purchase intention. So, we decided to make the book-buying process simpler with functions that make them control their order conveniently such as removing an item in their cart, removing all items in their cart, or increasing the quantity of the item.

Interpret question 6: The result of question 6 reveals that most people search for their desired books by the title of the book or the category of the book. Therefore, we will build functions for search books by title and category.

Interpret question 7: Most people think our system is working fine, but there are also some who think that the performance of the system doesn't work well and needs improvement. Interpret question 8: The results from question 8 have shown that the functions that customers want in a website:

- Extensive book selection
- User reviews and ratings
- Personalized book recommendations
- Easy navigation and search functionality
- Competitive pricing and discounts
- Ability to track orders and receive updates

In general, thanks to feedback, we discovered opportunities not previously considered such as the ability to track order function. So, in the next update, we will support all functions that customers expected from our system.

Interpret question 9: we gained a lot of useful recommendations from participants and we try to build a book-selling system that can meet the expectations and recommendations of our customers as below:

- Send a notification when users successfully make an order to ensure that their orders were processed
- Ensure that the website is mobile-friendly and has a responsive design that adjusts to different screen sizes.
- Make sure the search functionality is robust and allows users to easily find the books they're looking for by title, author, genre, or keyword.
- Provide a seamless and easy-to-use checkout process that includes multiple payment options.
- Offer fast and reliable shipping options with tracking and delivery notifications to keep users informed about the status of their orders.

4. Evaluate any new insights, ideas, or potential improvements

4.1 Extensive book selection

Customers are more likely to be satisfied with an e-commerce bookstore that offers a wide variety of books. By providing an extensive selection, customers are more likely to find the specific book rapidly they are looking for, which can lead to increased customer loyalty. Also, a larger selection of books can help an e-commerce bookstore improve its SEO. By providing a variety of books and categories, the bookstore can attract more organic search traffic from customers searching for specific titles, genres, or authors.

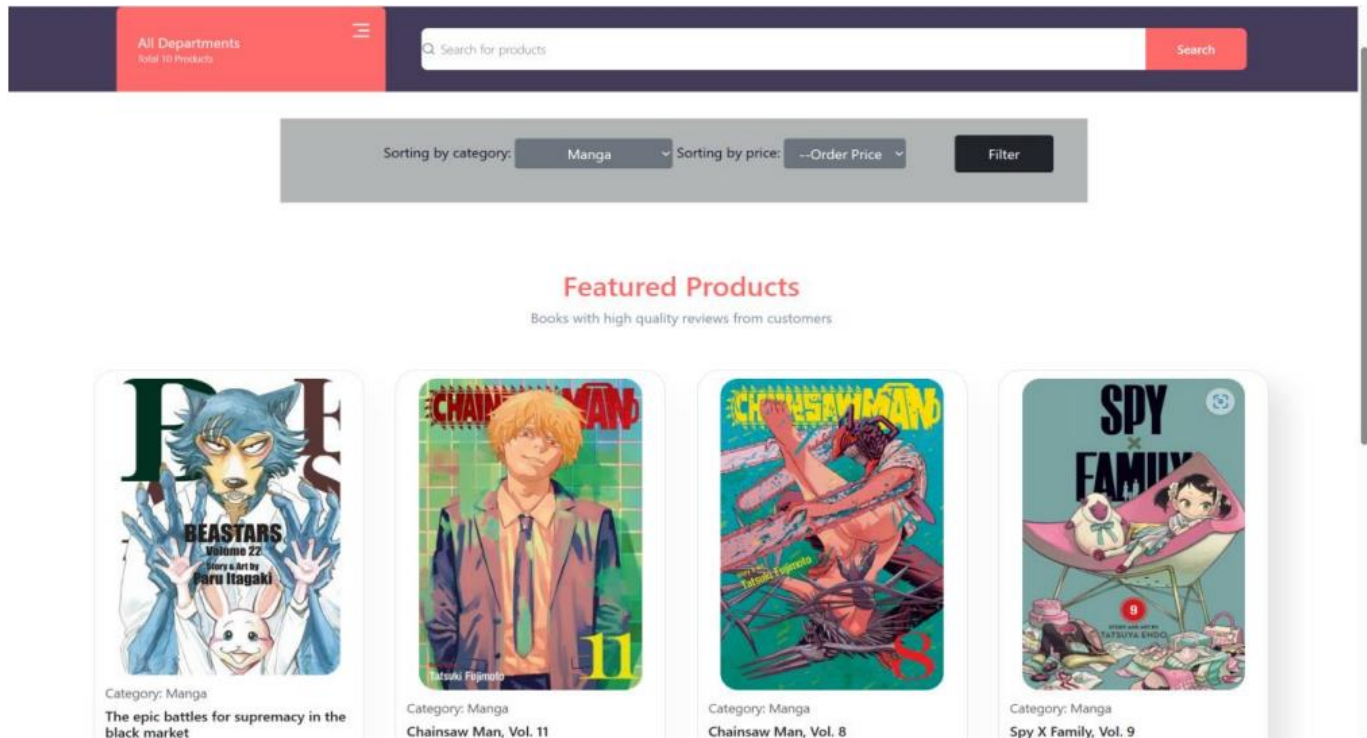


Figure 13: Book selection

4.2 Offering book recommendations

By offering personalized book recommendations, an e-commerce bookstore can improve the customer experience. Customers are more likely to be satisfied with a bookstore that offers recommendations that match their interests and reading habits. Moreover, customers who receive recommendations for books that match their interests are more likely to make additional purchases. This can help our shop to increase the profit.

Related Products



Figure 14: Offering book recommendations

4.3 Make the website to be responsive

A responsive website provides a better user experience for customers. It ensures that the website is accessible and easy to use on any device, whether it's a desktop computer, tablet, or mobile phone. This can lead to increased customer satisfaction and loyalty. In this part, we will use Bootstrap a popular front-end framework that provides a set of predesigned CSS components to build responsive websites and applications. Bootstrap provides a mobile-first approach to web design, meaning that the framework is optimized for smaller screens first and then scales up to larger screens.

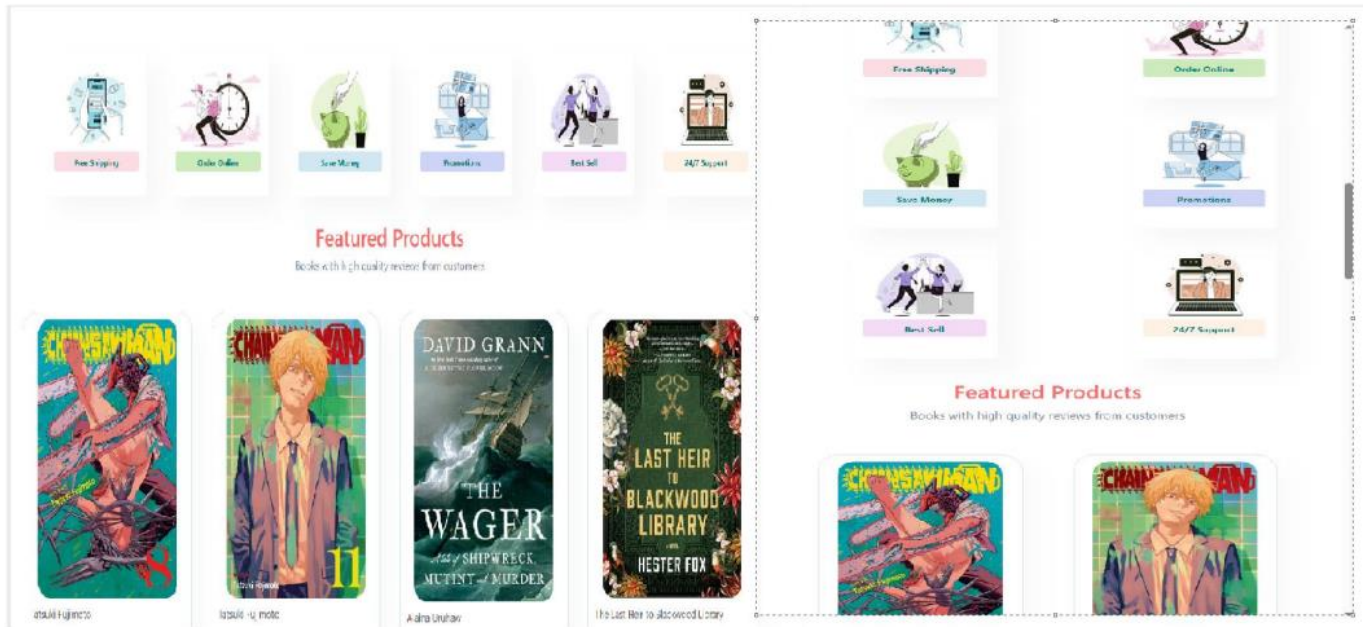


Figure 15: Responsive website

4.4 Send a notification when customers order

A notification confirms that the customer's order has been received and is being processed, which can help reduce anxiety and provide peace of mind. Moreover, customers appreciate being kept informed about their orders and are more likely to be satisfied with their overall shopping experience.

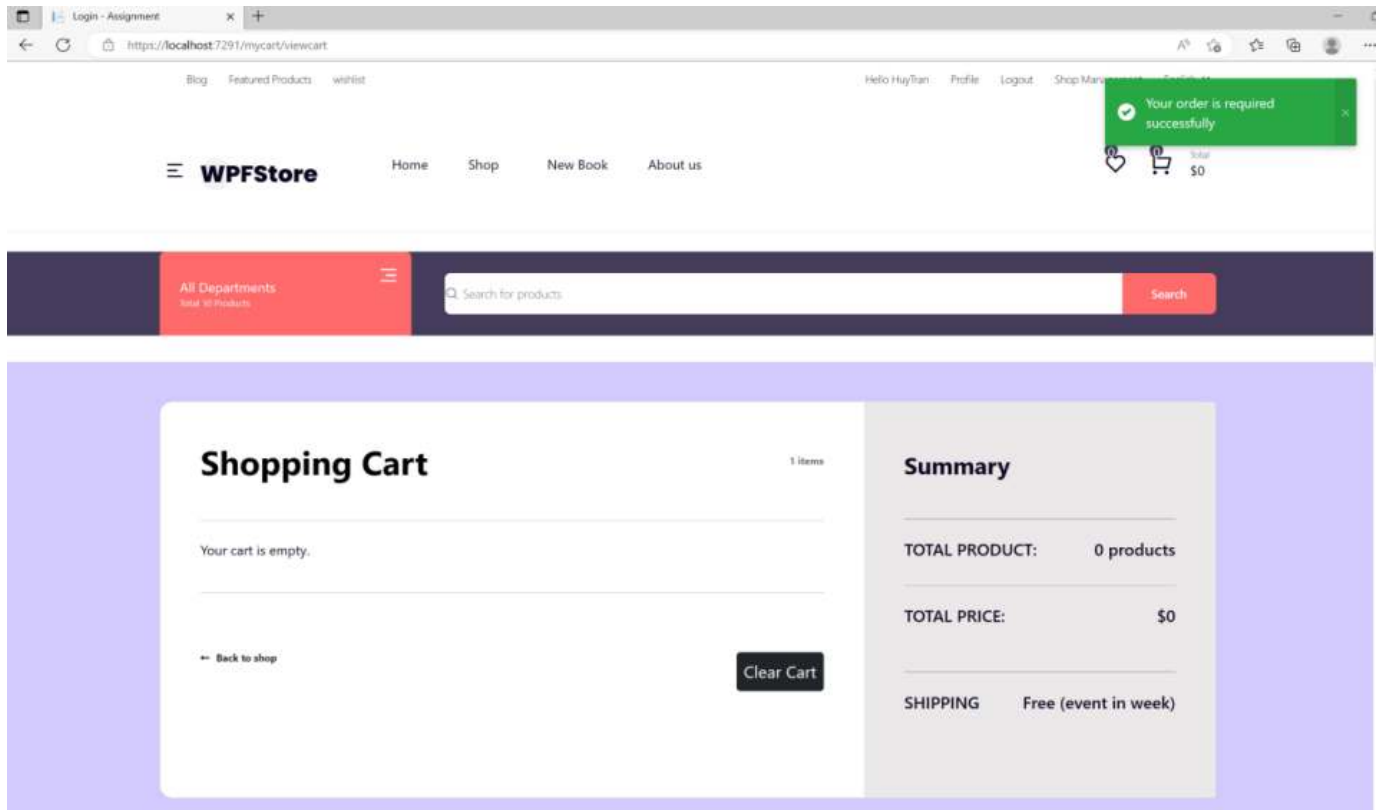


Figure 16: Sending notification

4.5 Show confirm notification before removing

Generally, a confirmation notification can help prevent the accidental deletion of important data. By requiring the user to confirm the action, it reduces the risk of unintentional deletion of data.


☰

🔍 Search projects

localhost:7291 says

Are you sure you want to remove?

OK Cancel

 Thien Nguyen

Request new cate

Search category 🔍

--Sorting by Status--

--Sorting by Time--

Filter

Category Table

ID	Category Name	Description	Status	Action
1	Action	for book that describes many action stories	Active	<div>Edit</div> <div>Remove</div>
2	Romantic	for book that describes many loved stories	Active	<div>Edit</div> <div>Remove</div>
3	Thrilling	for book that describes many thrilling stories	Active	<div>Edit</div> <div>Remove</div>
5	Manga	for book that describes cartoon stories from Japan	Active	<div>Edit</div> <div>Remove</div>

Page 1 of 2

Figure 17: Show confirmation

III / APPLICATION DEVELOPMENT (P5)

1. Folder structure of the application

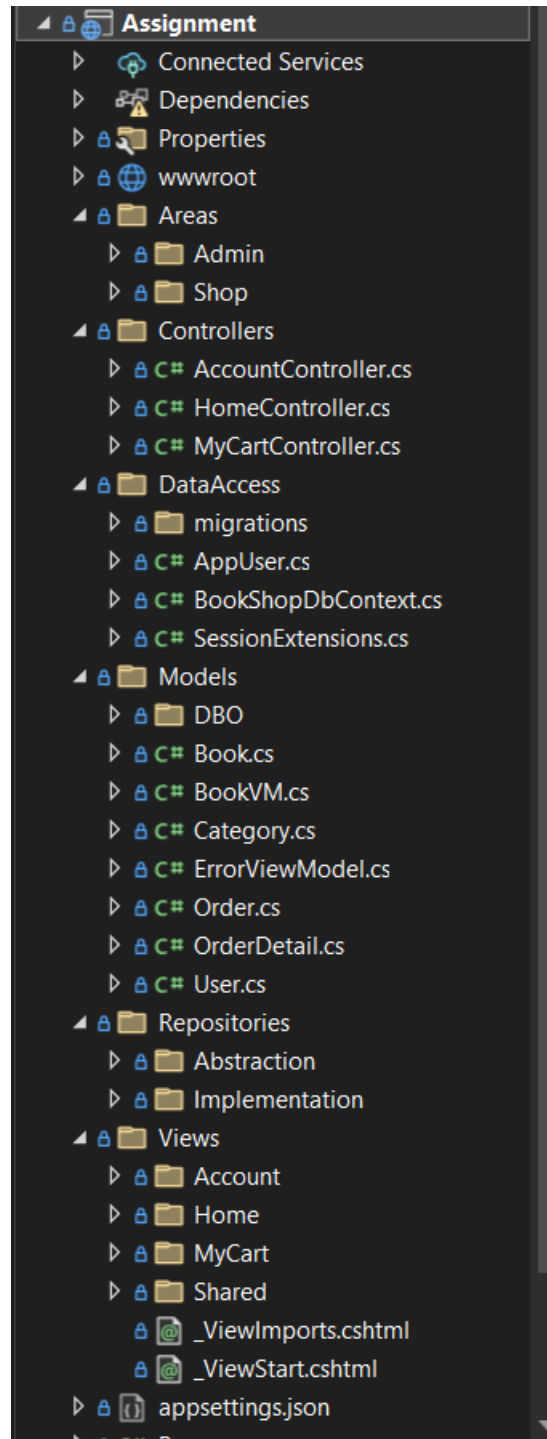


Figure 18: Folder structure of the application

ASP.NET Core is a popular web development framework that is widely used for building scalable and robust web applications. It follows the Model-View-Controller (MVC) pattern, which provides a clear separation of concerns between the different components of the application.

In our e-commerce application for selling books online, we have used ASP.NET Core to implement the platform. The layout structure of the application is quite easy to see, thanks to the MVC pattern. The application consists of several key folders, including Areas, DataAccess, Models, Controllers, and Views.

The Areas folder contains two roles, Admin and Shop, which are responsible for handling the different roles in the platform. The DataAccess folder is where the database is connected and where all the data access logic is stored. The Models folder contains all the objects that are stored in the database, including books, users, and orders.

The Controllers folder is where all the user's controllers are located. These controllers are responsible for handling incoming requests from users and returning the appropriate response. Finally, the Views folder contains the user interface, which is responsible for rendering the HTML that is displayed to users.

Overall, the use of ASP.NET Core in our e-commerce application for selling books online provides several key benefits, including scalability, robustness, and a clear separation of concerns. By following the MVC pattern and organizing the application into different folders and roles, we have created a platform that is easy to understand, maintain, and scale as needed.

My Contribution

As a member of a team of three, I have been chosen to build the shop functionality for our e-commerce application for selling books online. This is an exciting opportunity to work on a critical aspect of the platform and contribute to the successful delivery of a high-quality application.

My focus will be on building the functionality that allows users to browse, search, and purchase books on the platform. This will involve working closely with the other members of the team to ensure that my work is integrated seamlessly with the rest of the platform.

I will be responsible for ensuring that the shop functionality is easy to use, secure, and reliable. This will involve implementing robust data access logic, developing secure payment processing functionality, and ensuring that the user interface is intuitive and user-friendly.

Overall, I am excited to take on this role and contribute to the successful delivery of our e-commerce application for selling books online. By building a robust and user-friendly shop functionality, we aim to provide users with a seamless and enjoyable book shopping experience.

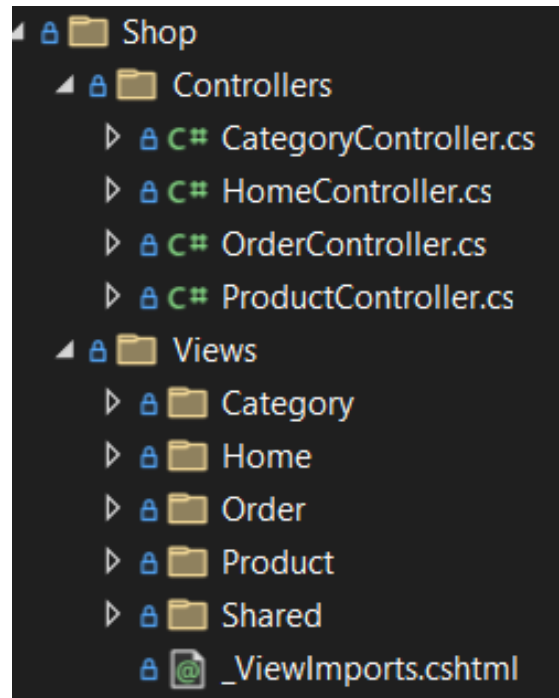


Figure 19: Shop role structure of the application

The shop section of our e-commerce application for selling books online will have a simple and intuitive structure, consisting of two main parts: the controller and the view.

The controller part will be divided into small tasks to handle, ensuring that each task is manageable and easy to maintain. This will involve breaking down the functionality of the shop section into smaller, more manageable components, such as browsing, searching, and purchasing books.

Similarly, the view will be designed with simplicity and user-friendliness in mind. This will involve creating a clear and intuitive user interface that is easy to navigate and understand, allowing users to browse, search, and purchase books with ease.

By breaking down the shop functionality into smaller, more manageable components and designing a simple and intuitive user interface, we aim to provide users with a seamless and enjoyable book shopping experience. This will help to ensure that users can find the books they need, make informed purchase decisions, and enjoy a hassle-free shopping experience on our e-commerce platform.

2. Source code samples of the application with explanation

2.1 Model

```
[Table("Book")]
23 references
public class Book
{
    [Key]
    [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
    20 references
    public int ID { set; get; }

    [Required(ErrorMessage = "Name cannot be empty !!")]
    [StringLength(50)]
    11 references
    public string Name { set; get; }

    [Column(TypeName = "Money")]
    [Required(ErrorMessage = "Price cannot be empty !!")]
    11 references
    public decimal Price { set; get; }
    [Column(TypeName = "Money")]
    9 references
    public decimal DiscountPrice { set; get; }

    [StringLength(250)]
    [Required(ErrorMessage = "Description cannot be empty !!")]
    // ...
}
```

Figure 20: Book class

The provided code shows the Book class, which is a model for storing information about books in our e-commerce application for selling books online. It includes properties for the book's ID, name, price, discount price, description, quantity, image, author, status, and category. The CategoryID property is a foreign key that links to the Category class, which stores information about book categories. By using this model, we can store and retrieve information about books in our application's database, allowing users to browse, search, and purchase books on the platform.

```
public class Category
{
    [Key]
    [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
    14 references
    public int ID { set; get; }

    [Required(ErrorMessage = "Name cannot be empty !!")]
    [StringLength(50)]
    10 references
    public string Name { set; get; }

    [StringLength(50)]
    6 references
    public string? Alias { set; get; }

    [Required(ErrorMessage = "Status cannot be empty !!")]
    [StringLength(50)]
    14 references
    public string Status { set; get; } = "Processing";

    [StringLength(250)]
    8 references
    public string? Description { set; get; }
```

Figure 21: Category class

The provided code shows the Category class, which is a model for storing information about book categories in our e-commerce application for selling books online. It includes properties for the category's ID, name, alias, status, and description. The Books property is a virtual list of Book objects, which allows us to store information about the books in each category. We can use this model to store and retrieve information about book categories in our application's database, allowing users to browse and search books by category on the platform. By using this model, we can organize the books on our platform into different categories, making it easier for users to find the books they're interested in.

```
public class Order
{
    [Key]
    [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
    11 references
    public int ID { set; get; }
    [Required]
    5 references
    public string UserID { set; get; }

    [Column(TypeName = "Money")]
    9 references
    public decimal? Total { set; get; }

    [Required]
    12 references
    public DateTime OrderDate { set; get; }

    [Required]
    [StringLength(50)]
    8 references
    public string Status { set; get; }

    [StringLength(50)]
    7 references
    public string PaymentMethod { set; get; }
```

Figure 22: Order class

The provided code shows the Order class, which is a model for storing information about orders in our e-commerce application for selling books online. It includes properties for the order's ID, user ID, total, order date, status, payment method, delete flag, and note. The User property is a foreign key that links to the AppUser class, which stores information about application users. The OrderDetails property is a virtual list of OrderDetail objects, which allows us to store information about the books in each order. We can use this model to store and retrieve information about orders in our application's database, allowing users to view their order history and track the status of their orders. By using this model, we can provide users with a seamless and enjoyable shopping experience on our e-commerce platform.

2.2 View

```
<div class="content-wrapper">
  <div class="row">
    <div class="d-flex mb-3">
      <div class="me-auto p-2">
        <form asp-area="Shop" asp-controller="Category" asp-action="Index">
          <div class="input-group">
            <div class="form-outline">
              <input type="search" name="search" id="form1" class="form-control" placeholder="Search category" />
            </div>
            <button type="submit" class="btn btn-primary">
              <i class="fas fa-search"></i>
            </button>
          </div>
        </form>
      </div>
      <div class="p-2">
        <form asp-area="Shop" asp-controller="Category" asp-action="AddCategory" method="post">
          <button type="submit" class="btn btn-primary">Request new category</button>
        </form>
      </div>
    </div>
  </div>

  <div class="row">
    <div class="sidebar-actions">
      @using (Html.BeginForm("Index", "Category", FormMethod.Get))
      {
        <p style="align-content:center">
          @Html.DropDownList("status", (SelectList)ViewBag.StatusList, "--Sorting by Status--", new { @class = "btn btn-danger dropdown-toggle text-center" })
          @Html.DropDownList("sortNew", (SelectList)ViewBag.NewSort, "--Sorting by Time--", new { @class = "btn btn-danger dropdown-toggle" })
          <input type="submit" class="btn btn-gradient-primary" value="Filter" />
        </p>
      }
    </div>
  </div>

  <div class="row">
    <div class="col-lg-12 grid-margin stretch-card">
```

Figure 23: Category (index)

This code represents the front-end part of a web page for managing categories in an e-commerce application. It includes a search box and a "Request new category" button. The page also allows users to filter and sort the category table according to status and time. The category table displays the categories' ID, name, description, and status. Users can edit or remove categories using the corresponding buttons in the action column. The page also includes a pagination component to allow users to navigate through the different pages of the category table. Overall, this code provides a user-friendly interface for managing categories in an e-commerce application.

```


<div class="content-wrapper">
    <div class="page-header">
      <h3 class="page-title">
        <span class="page-title-icon bg-gradient-primary text-white me-2">
          <i class="fa-solid fa-house"></i>
        </span> Dashboard
      </h3>
      <nav aria-label="breadcrumb">
        <ul class="breadcrumb">
          <li class="breadcrumb-item active" aria-current="page">
            <span></span>Overview <i class="mdi mdi-alert-circle-outline icon-sm text-primary align-middle"></i>
          </li>
        </ul>
      </nav>
    </div>
    <div class="row">
      <div class="col-md-4 stretch-card grid-margin">
        <div class="card bg-gradient-primary card-img-holder text-white">
          <div class="card-body">
            
            <h4 class="font-weight-normal mb-3">
              Total Sales <i class="mdi mdi-chart-line mdi-24px float-right"></i>
            </h4>
            <h2 class="mb-5">$ @ViewBag.TotalSale.ToString("0.##")</h2>
          </div>
        </div>
      </div>
    </div>
  </div>


```

Figure 24: Home (index)

The given code appears to be a section of a web page's HTML markup, specifically a dashboard page. It contains six cards that display various metrics related to the website's sales, orders, products, and users. The values for these metrics are dynamically generated using server-side code to fetch data from a database and display them on the webpage. The cards are styled using Bootstrap's CSS classes to give them a gradient background and an image overlay. The use of icons from the Material Design Icon Library enhances the visual appeal of the dashboard.

```

@*<div class="main-panel">*@
  <div class="content-wrapper">
    <div class="row">
      <div class="d-flex mb-3">
        <div class="me-auto p-2">
          <div class="input-group">
            <div class="form-outline">
              <input type="search" id="form1" class="form-control" placeholder="Search product" />
            </div>
            <button type="button" class="btn btn-primary">
              <i class="fas fa-search"></i>
            </button>
          </div>
        </div>
      </div>
    </div>
    <div class="row">
      <div class="sidebar-actions">
        <using (Html.BeginForm("Index", "Order", FormMethod.Get))>
        {
          <p style="align-content:center">
            @Html.DropDownList("status", (SelectList)ViewBag.StatusList, "--Sorting by Status--", new { @class = "btn btn-danger dropdown-
            @Html.DropDownList("sortNew", (SelectList)ViewBag.NewSort, "--Sorting by Time--", new { @class = "btn btn-danger dropdown-togg
            <input type="submit" class="btn btn-gradient-primary" value="Filter" />
          }
        }
      </div>
    </div>
  </div>

```

Figure 25: Order (index)

The given code appears to be a section of a web page's HTML markup for a dashboard displaying a table of orders. It includes a search bar for searching products, a form with dropdown menus for sorting and

filtering orders, and a table displaying order details such as ID, order date, status, total, and payment method. The table also includes a button for viewing order details and a form for removing orders.

```
<div class="row">
  <div class="sidebar-actions">
    @using (Html.BeginForm("Index", "Product", FormMethod.Get))
    {
      <p style="align-content:center">
        @Html.DropDownList("status", (SelectList)ViewBag.StatusList, "--Sorting by Status--", new { @class = "btn btn-danger dropdown-toggl
        @Html.DropDownList("sortPrice", (SelectList)ViewBag.PriceSort, "--Sorting by Price--", new { @class = "btn btn-danger dropdown-togg
        <input type="submit" class="btn btn-gradient-primary" value="Filter" />
      </p>
    }
  </div>
</div>

<div class="row">
  <div class="col-lg-12 grid-margin stretch-card">
    <div class="card">
      <div class="card-body">
        <h4 class="card-title">Book Table</h4>

        <table class="table table-striped" style="width:100%;">
          <thead>
            <tr>
              <th> ID </th>
              <th> Product Name </th>
              <th> Image </th>
              <th> Price </th>
              <th> Quantity </th>
            </tr>
          </thead>
        </table>
      </div>
    </div>
  </div>
</div>
```

Figure 26: Product (index)

The given code is a section of a web page's HTML markup for an online bookstore. It includes a form with dropdown menus for sorting and filtering products, and a table displaying product details such as ID, product name, image, price, quantity, and status. The table also includes a button for managing the product's status. The code is likely part of a larger web application for managing products and sales in an online bookstore. Overall, this code provides a user-friendly interface for customers to search and filter products based on price and status.

2.3 Controller

```
[HttpPost]
[ValidateAntiForgeryToken]
0 references
public async Task<IActionResult> AddCategory([Bind("ID,Name,Description,Alias,Status")]Category category)
{
    if(_bookShopDbContext.Category.Any(c => c.Name == category.Name))
    {
        return Json("This category already existed");
    }
    else
    {
        if (ModelState.IsValid)
        {
            _bookShopDbContext.Category.Add(category);
            await _bookShopDbContext.SaveChangesAsync();
            _notyfService.Success("You successfully create a new category");
            return RedirectToAction("index");
        }
    }
    return View();
}
```

Figure 27: Category controller

The given code is a C# method for adding a new category to an online bookstore. It receives a Category object as a parameter, which contains properties such as ID, Name, Description, Alias, and Status. The method first checks if a category with the same name already exists in the database, and returns an error message if so. If the category is new and passes validation, it is added to the database using Entity Framework, and a success message is displayed using the NotyfService library. Finally, the method redirects the user to the index page displaying all categories.

```
0 references
public IActionResult Index()
{
    var now = DateTime.Now.Month;
    ViewBag.TotalSale = _bookShopDbContext.Order.Sum(i => i.Total);
    ViewBag.TotalOrder = _bookShopDbContext.Order.Count();
    ViewBag.TotalProduct = _bookShopDbContext.Book.Count();
    ViewBag.SaleMonth = _bookShopDbContext.Order.Where(i => i.OrderDate.Month == now).Sum(i => i.Total);
    ViewBag.OrderMonth = _bookShopDbContext.Order.Where(i => i.OrderDate.Month == now).Count();
    ViewBag.Users = _bookShopDbContext.AppUser.Count();

    ViewBag.RecentOrders = _bookShopDbContext.Order.OrderBy(i => i.OrderDate).Take(4).ToList();
    ViewBag.InforUsers = _bookShopDbContext.AppUser.ToList();
    ViewBag.TopProducts = _bookShopDbContext.OrderDetail
        .Join(_bookShopDbContext.Book, od => od.ProductID, p => p.ID, (od, p) => new { OrderDetail = od, Book = p })
        .GroupBy(od => od.OrderDetail.ProductID)
        .OrderByDescending(q => q.Sum(o => o.OrderDetail.Quantity))
        .Take(3).Select(g => g.FirstOrDefault().Book);
    ViewBag.Books = _bookShopDbContext.Book.ToList();
    _notyfService.Information("Welcome to the Shop Owner board");
    return View();
}
```

Figure 28: Home controller

The given code is a C# method for displaying an overview dashboard of an online bookstore. It uses Entity Framework and LINQ queries to retrieve information from the database and set various ViewBag properties.

The first few ViewBag properties calculate and display the total sales, orders, and products, as well as the sales and orders for the current month. The next few ViewBag properties display information about recent orders, user information, and the top-selling products.

The final ViewBag property displays all available books in the bookstore. Additionally, the NotyfService library is used to display a welcome message to the user.

Overall, this code provides a quick overview of the current state of the bookstore's business, including sales, orders, user information, and top-selling products.

```
public async Task<IActionResult> ViewDetail(int id)
{
    var order = _bookShopDbContext.Order.FirstOrDefault(o => o.ID == id);
    var customer = _bookShopDbContext.AppUser.FirstOrDefault(c => c.Id == order.UserID);
    var ListProduct = _bookShopDbContext.OrderDetail.Where(od => od.OrderID == order.ID).ToList();
    ViewBag.ListProducts = ListProduct;
    ViewBag.InforCustomer = customer;
    ViewBag.Books = _bookShopDbContext.Book.ToList();

    OrderViewModel viewModel = new OrderViewModel
    {
        ID = order.ID,
        UserID = order.UserID,
        Total = order.Total,
        Delete = order.Delete,
        Note = order.Note,
        OrderDate = order.OrderDate,
        PaymentMethod = order.PaymentMethod,
        Status = order.Status
    };

    List<SelectListItem> items = new List<SelectListItem>();
    items.Add(new SelectListItem { Text = "Pending", Value = "Pending" });
    items.Add(new SelectListItem { Text = "Processing", Value = "Processing" });
    items.Add(new SelectListItem { Text = "Delivered", Value = "Delivered" });
    items.Add(new SelectListItem { Text = "Cancelled", Value = "Cancelled" });
    ViewBag.StatusList = new SelectList(items, "Text", "Value");
    return View(order);
}
```

Figure 29: Order controller

It receives an ID parameter that represents the ID of the order to be displayed. The method retrieves information about the order, including the customer who placed the order and a list of products included in the order, using Entity Framework and LINQ queries. It sets various ViewBag properties to display the list of products, customer information, and available books.

The method also creates an instance of the OrderViewModel class, which contains properties for the order's ID, user ID, total, delete status, note, order date, payment method, and status. Additionally, the

method creates a SelectList object containing options for the order's status, such as pending, processing, delivered, and cancelled.

Overall, this code provides a detailed view of a specific order in an online bookstore, including the customer who placed the order, the products included in the order, and the order's status.

```

List<Category> SuitableCategories = _bookShopDbContext.Category.Where(c => c.Status.Equals("Active")).ToList();
ViewBag.categoryList = new SelectList(SuitableCategories, "ID", "Name");
return View(bookVM);
}

[HttpPost]
[ValidateAntiForgeryToken]
0 references
public ActionResult RemoveConfirmed(int id)
{
    Book book = _bookShopDbContext.Book.FirstOrDefault(c => c.ID == id);
    if (id == null || book == null)
    {
        return View("Error");
    }
    _bookShopDbContext.Book.Remove(book);
    _bookShopDbContext.SaveChanges();
    return RedirectToAction("Index");
}

```

Figure 30: Product controller

This code will check if the book exists and if it does, it will be removed from the store. The code also contains an error check to make sure that it is not trying to remove a book that does not exist. This code will help ensure that customers are not able to purchase books that are no longer available.

3. Final screenshots of the application

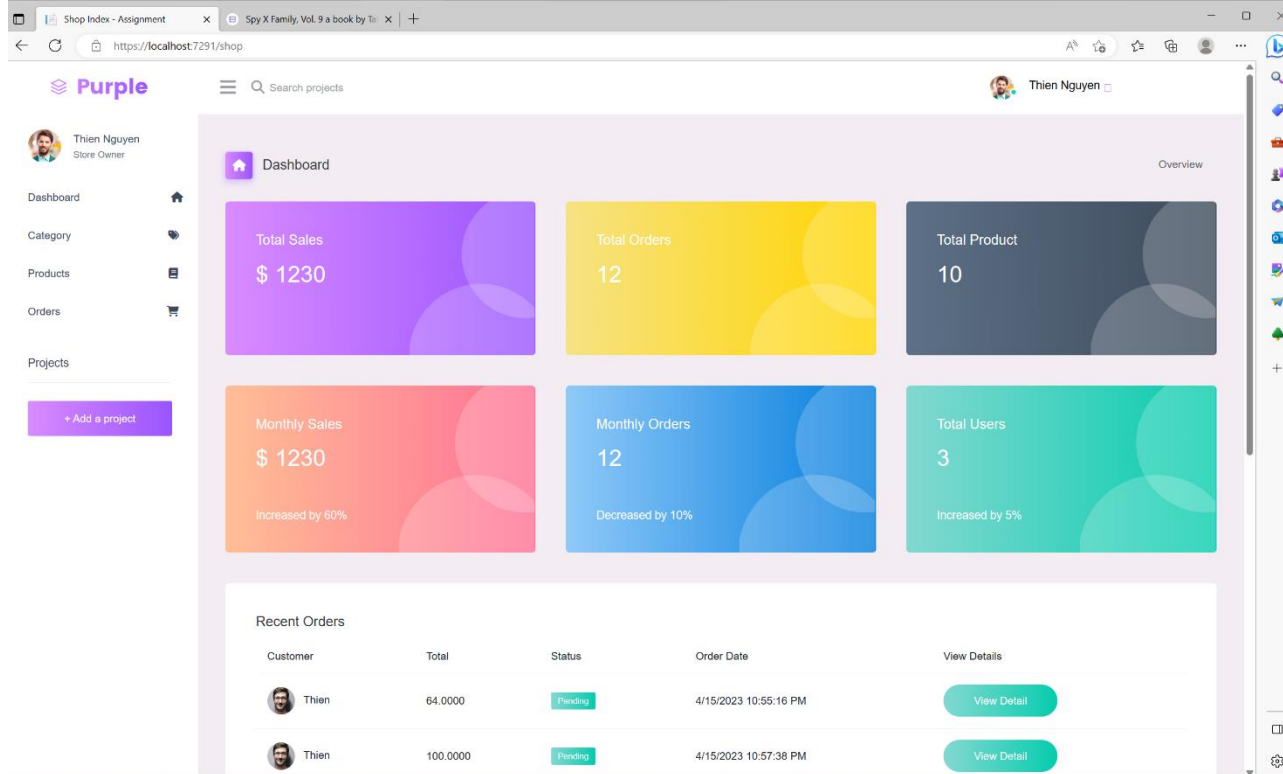


Figure 31: Dashboard

To make the Dashboard page even better, the interface could be made more visually appealing with modern design elements like flat design, bold typography, and a cohesive color scheme. Additionally, incorporating interactive features like charts, graphs, and data visualization tools could help the shop owner quickly understand the business's current state. Search and filtering options, quick links to frequently used functions, and customizable settings could also improve the page's user-friendliness. Continual updates and improvements to the Dashboard page are crucial to the success of an online bookstore.

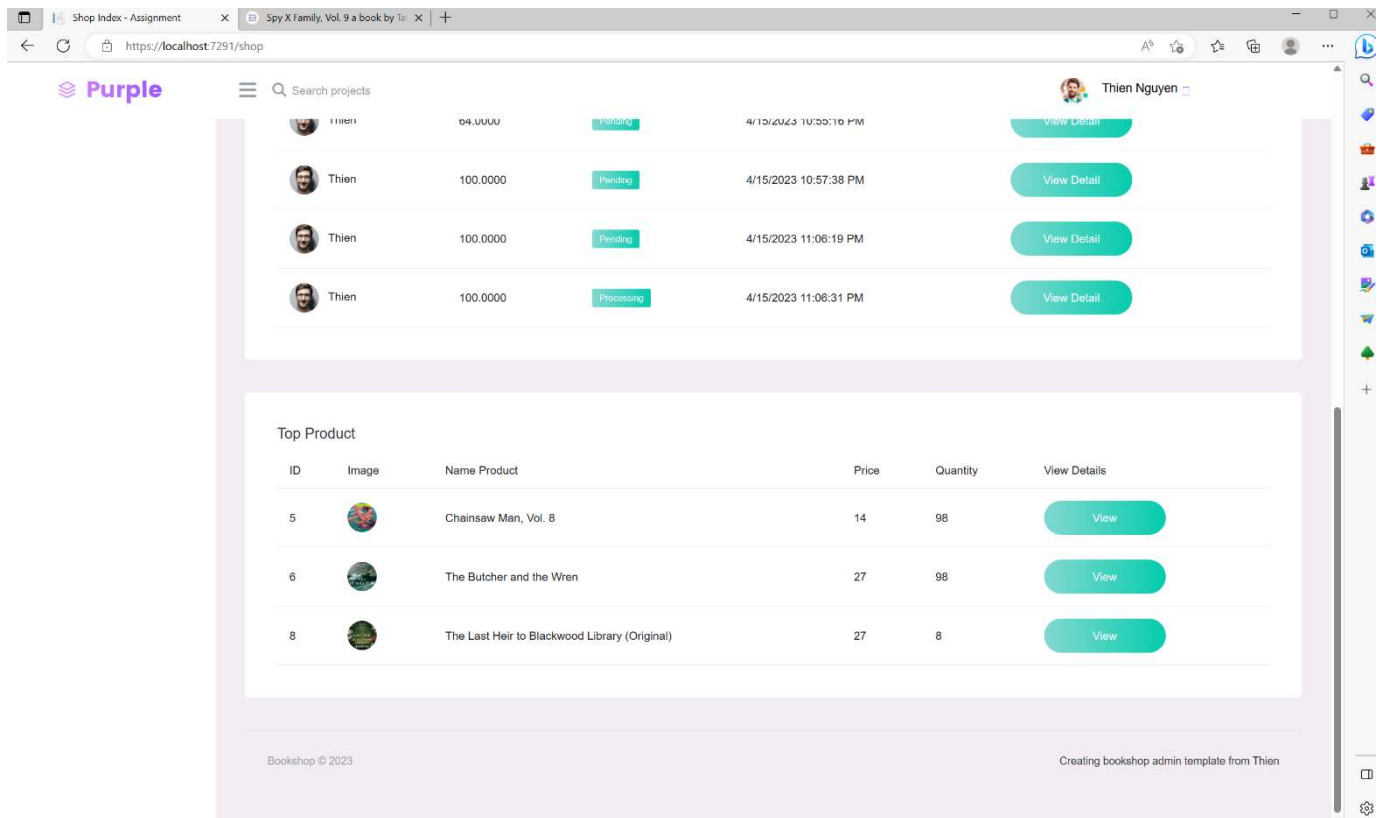


Figure 32: List order and top product

To enhance the readability and user-friendliness of the order and product lists, several improvements could be made. For example, the data could be sorted and filtered by various criteria, such as date, customer, product category, or sales volume. Additionally, the table could be made more interactive by allowing the user to click on a row to view more details about the order or product. To improve the visual appeal of the table, alternating row colors or custom formatting could be used. Overall, these enhancements could help the shop owner quickly and easily access the information they need to manage their online bookstore effectively.

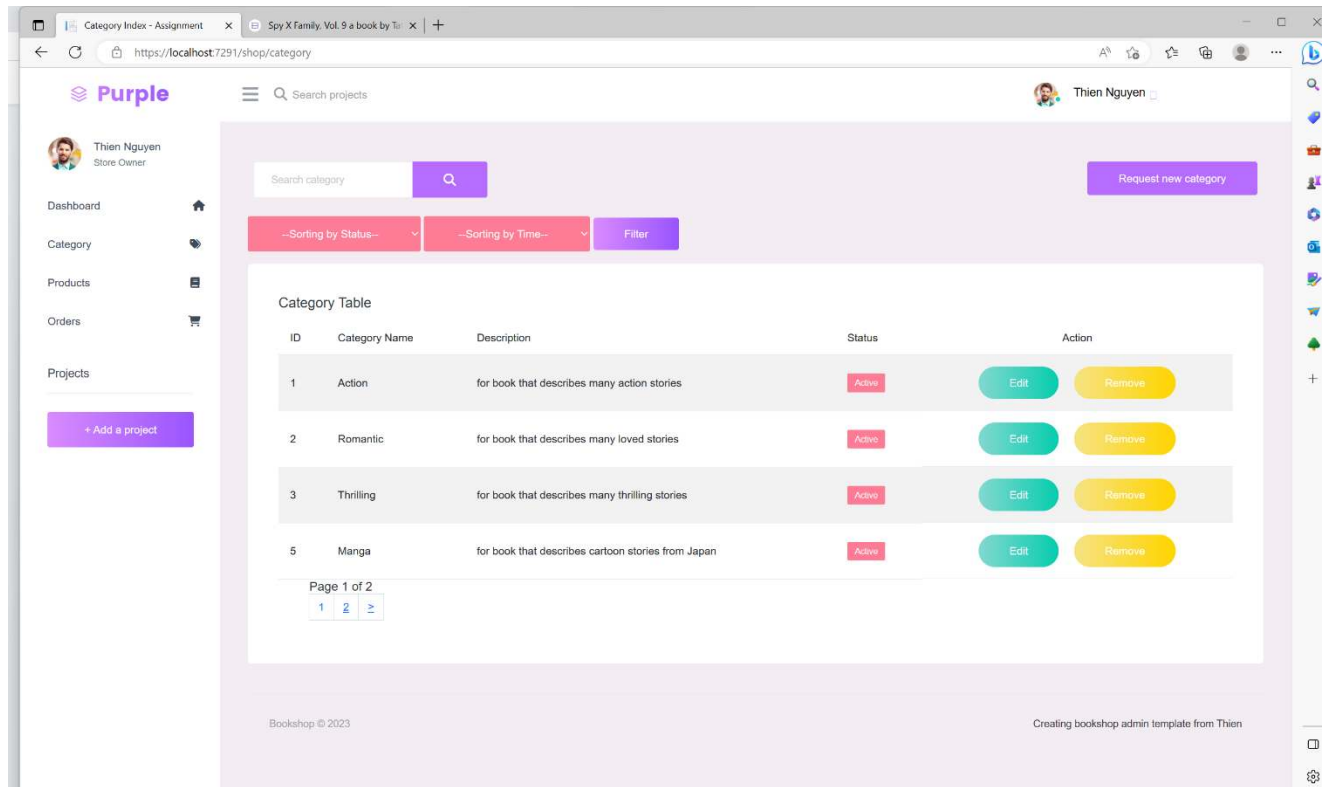
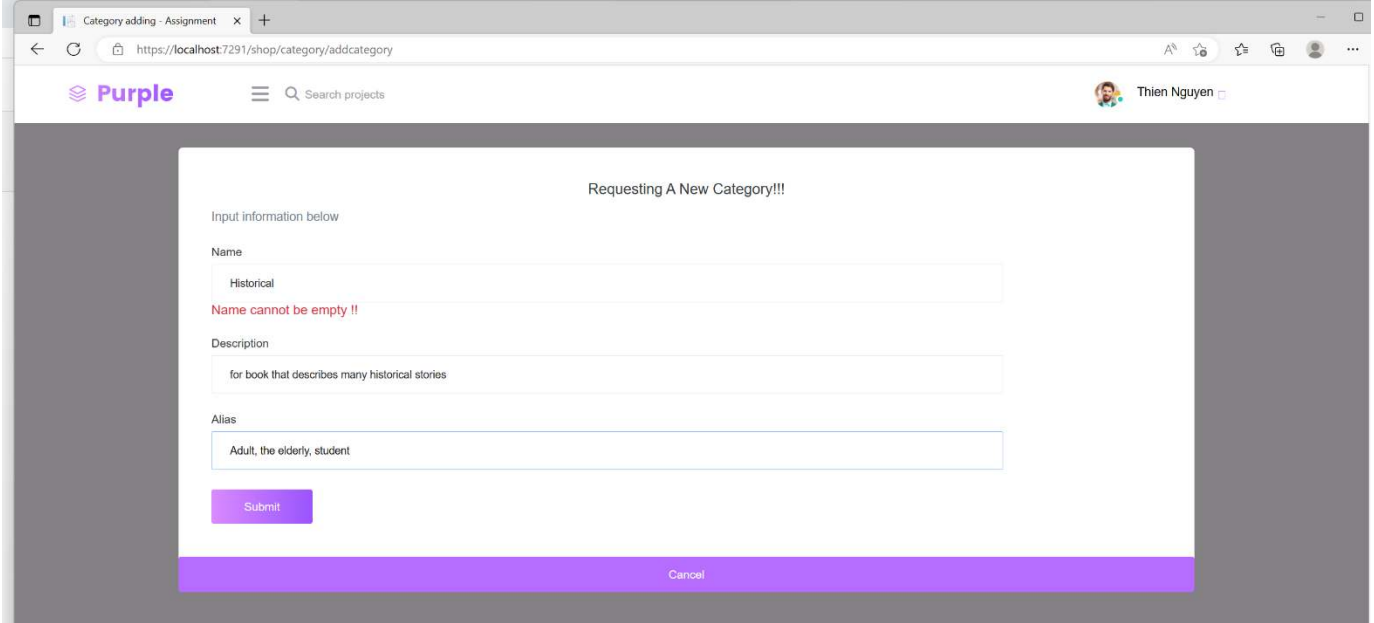


Figure 33: Manage category

The interface for managing categories in an online shop is an essential feature, and several enhancements can be made to improve its usability. One way to enhance the interface is to organize categories into a hierarchical structure, allowing for easier navigation between parent and child categories. Adding search and filter functionality can help users quickly find specific categories based on name, description, or status. Interactive features such as drag-and-drop functionality can improve the user experience by making it easier to reorder categories or create new ones. Finally, to improve accessibility, keyboard shortcuts or screen reader support can be added. Overall, these enhancements would make it easier for shop owners to manage their categories efficiently and provide a better user experience for shop role.



Category adding - Assignment x +

https://localhost:7291/shop/category/addcategory

Purple Search projects Thien Nguyen

Requesting A New Category!!!

Input information below

Name

Historical

Name cannot be empty !!

Description

for book that describes many historical stories

Alias

Adult, the elderly, student

Submit

Cancel

Figure 34: Request add new category

The interface for adding a new category in an online shop is a critical feature, and several enhancements could be made to improve its usability. The interface typically includes fields for Name, Description, and Alias, which can be filled in by the shop owner. To enhance the user experience, the form could be made more intuitive and user-friendly by including input validation and error messages. Additionally, the form could be made more interactive by providing suggestions for category names or descriptions based on previous entries. Finally, to improve accessibility, the form should be designed with consideration for users of all abilities, including those who may use screen readers or other assistive technologies. Overall, these enhancements would make it easier for shop owners to add new categories and improve the overall user experience for their customers.

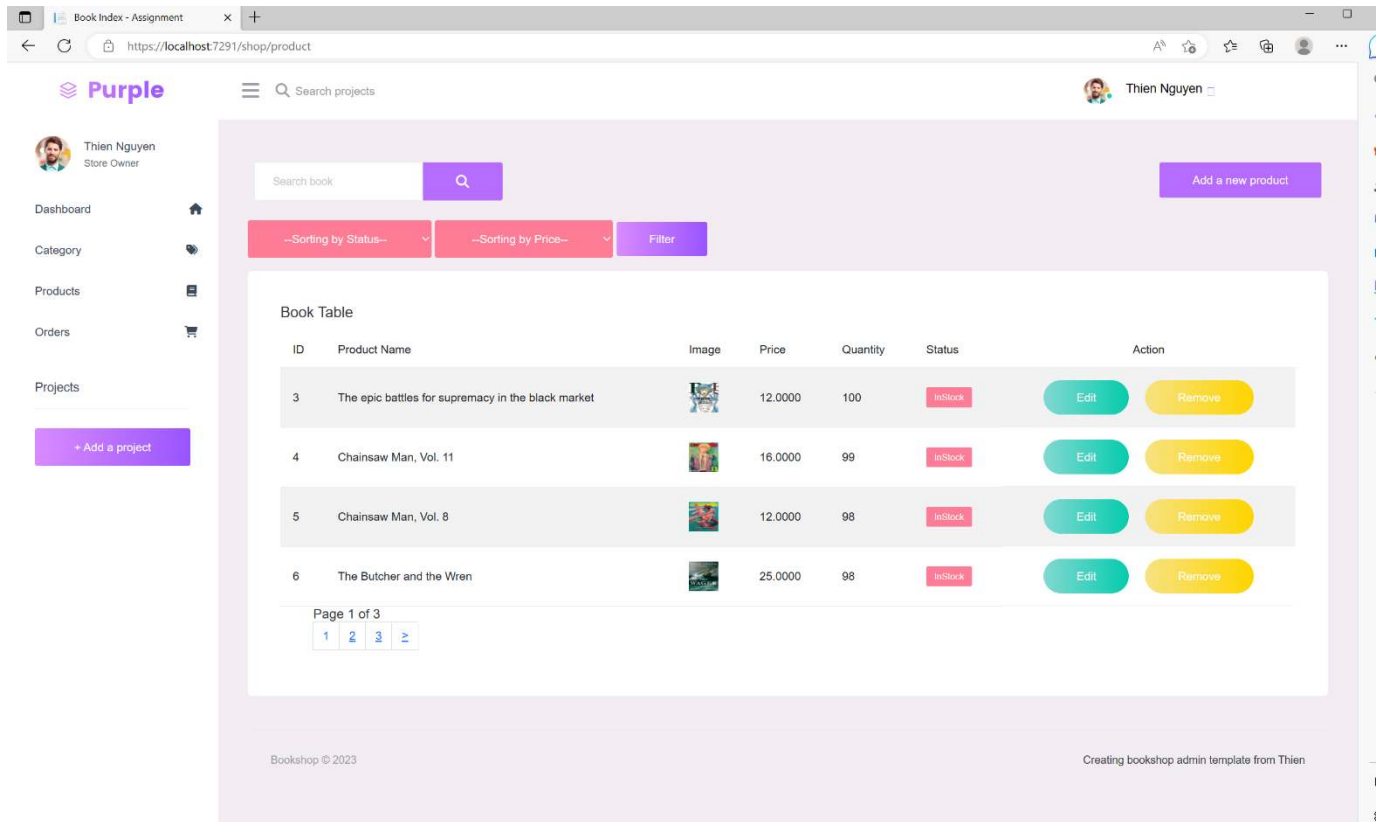
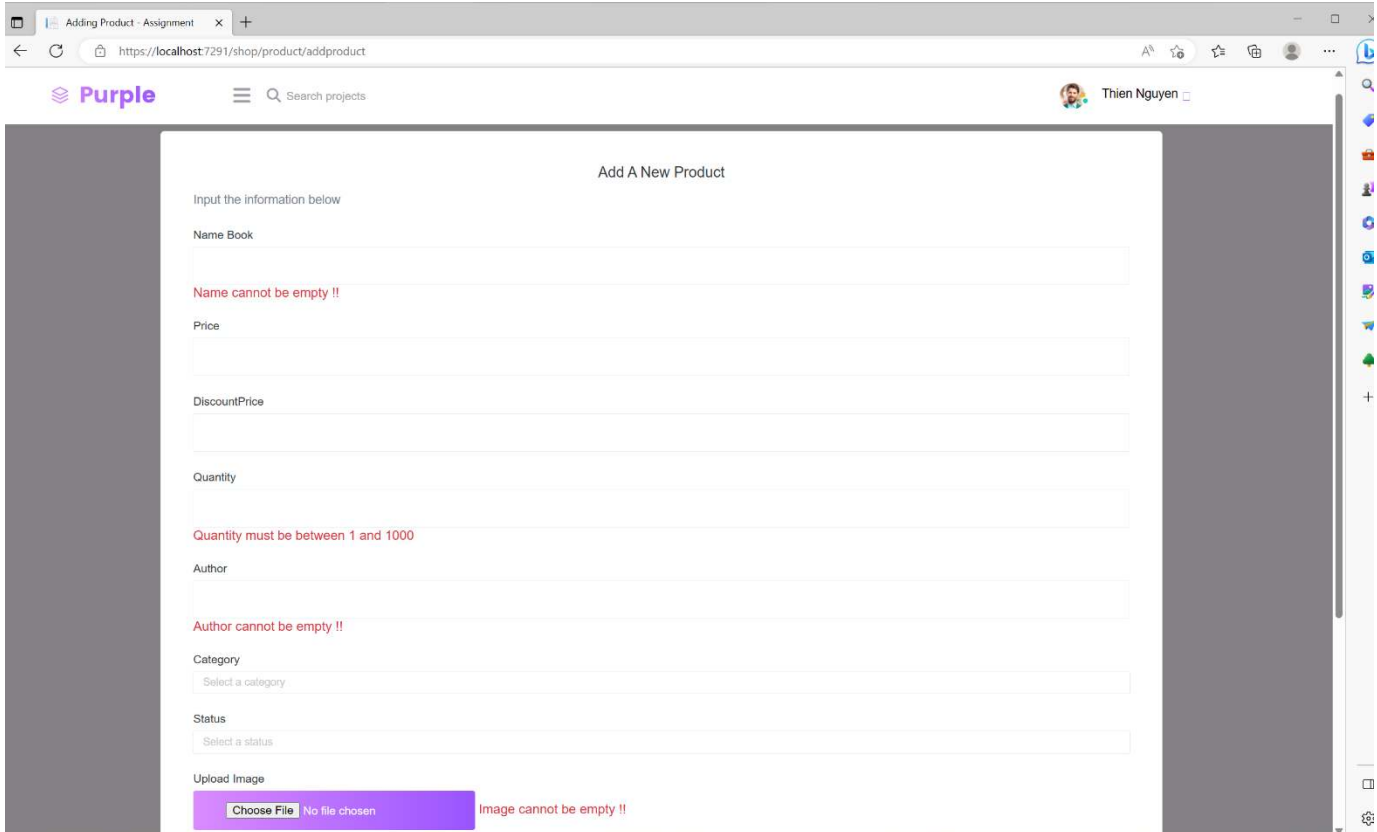


Figure 35: Manage book

The interface for managing books in an online shop is a crucial feature, and several enhancements could be made to improve its usability. The interface typically includes fields for Name, Image, Price, Quantity, and Status, presented in a tabular form that is visually appealing and easy to read. To enhance the user experience, the table could be made more interactive by allowing the user to click on a row to view more details about the book or perform actions such as editing or deleting the book. Additionally, search and filter functionality could be added to help the user quickly find specific books based on criteria such as author, category, or price range. Finally, to improve accessibility, the interface should be designed with consideration for users of all abilities, including those who may use screen readers or other assistive technologies. Overall, these enhancements would make it easier for shop owners to manage their books and improve the overall user experience for their customers.



Adding Product - Assignment

https://localhost:7291/shop/product/addproduct

Purple Search projects Thien Nguyen

Add A New Product

Input the information below

Name Book

Name cannot be empty !!

Price

DiscountPrice

Quantity

Quantity must be between 1 and 1000

Author

Author cannot be empty !!

Category

Select a category

Status

Select a status

Upload Image

Choose File No file chosen Image cannot be empty !!

Figure 36: Add a new book

The interface for adding a new book to an online shop is a crucial feature, and several enhancements could be made to improve its usability. The interface typically includes fields for Name, Image, Price, Quantity, and Status, which the shop owner fills in to post the book on the website. To enhance the user experience, the form could be made more intuitive and user-friendly by including input validation and error messages. Additionally, the form could be made more interactive by providing suggestions for book names or descriptions based on previous entries. To improve the quality of book listings, shops could also incorporate features such as spell-checking and recommended image sizes. Finally, the form should be designed with consideration for users of all abilities, including those who may use screen readers or other assistive technologies. Overall, these enhancements would make it easier for shop owners to add new books and improve the overall user experience for their customers.

The screenshot shows a web browser window with the URL `https://localhost:7291/shop/order/viewdetail/16`. The page has a purple header with a logo and a search bar. The main content area is divided into two sections: "Information Order" and "Information Customer".

Information Order

Update status below

Total Money	98.0000
Order Date	4/19/2023 11:05:26 AM
Payment Method	Visa
Note	hehe
Category	Pending

Save

Information Customer

Full Name	Thien
Phone Number	123456789

Figure 37: Customer detail and order

The interface for managing customer orders in an online shop is a crucial feature, and several enhancements could be made to improve its usability. The interface typically includes fields for order ID, customer name, order date, and order status, presented in a tabular form that is visually appealing and easy to read. To enhance the user experience, the table could be made more interactive by allowing the user to click on a row to view more details about the order or perform actions such as updating the order status or contacting the customer. Additionally, search and filter functionality could be added to help the user quickly find specific orders based on criteria such as order ID, customer name, or order status. Finally, the interface should be designed with consideration for users of all abilities, including those who may use screen readers or other assistive technologies. Overall, these enhancements would make it easier for shop owners to manage customer orders and improve the overall user experience for their customers.

[Back](#)

Book Table

ID Product	Name Product	Price	Quantity	Total	Action
6	The Butcher and the Wren	\$25.0000	2	\$50.0000	View Detail
8	The Last Heir to Blackwood Library (Original)	\$24.0000	2	\$48.0000	View Detail

Copyright © bootstrapdash.com 2021
Free [Bootstrap admin template](#) from Bootstrapdash.com

Figure 38: Other order

The interface for managing orders in an online shop is a critical feature, and several enhancements could be made to improve its usability. The interface typically includes a list of orders, presented in a tabular form that is visually appealing and easy to read. To enhance the user experience, the table could be made more interactive by allowing the user to click on a row to view more details about the order or perform actions such as editing or cancelling the order. The interface could also incorporate features such as order tracking, which would allow the customer to monitor the status of their order in real-time. Finally, to improve accessibility, the interface should be designed with consideration for users of all abilities, including those who may use screen readers or other assistive technologies. Overall, these enhancements would make it easier for shop owners to manage orders and improve the overall user experience for their customers.

4. Screenshots of using GitHub to manage the source code

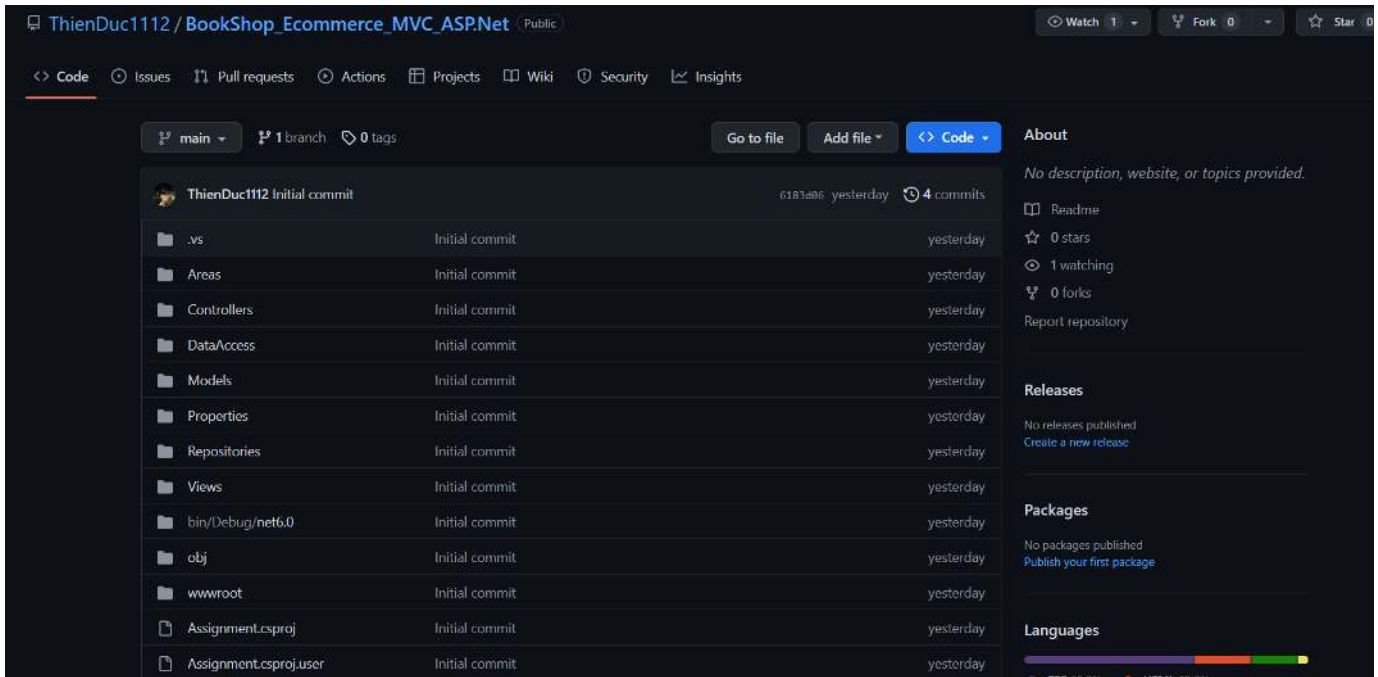


Figure 39: Using GitHub to manage the source code

GitHub is a powerful tool for managing source code in a team environment. It offers several features that make it an effective platform for collaboration, such as version control, issue tracking, and code review. With version control, team members can work on the same codebase simultaneously, and changes can be tracked and merged seamlessly. Issue tracking allows team members to report bugs or feature requests and assign them to specific team members for resolution. Additionally, code review allows team members to review and provide feedback on each other's code, ensuring that the codebase is of high quality and consistent with best practices. Finally, GitHub's integration with other tools, such as continuous integration and deployment services, makes it a powerful platform for managing the entire software development lifecycle. Overall, using GitHub can significantly increase the productivity and effectiveness of a team in managing their source code.

5. Screenshots of using IIS or Azure for the application deployment

For the deployment of our application, my team decided to select Azure, a cloud computing platform offered by Microsoft.

Step 1: Create a new web app on the Azure website

Home > Create a resource >

Create Web App ...

Instance Details

Need a database? Try the new Web + Database experience. [🔗](#)

Name * .azurewebsites.net

Publish * ☒ Code ☐ Docker Container ☐ Static Web App

Runtime stack *

Operating System * ☐ Linux ☒ Windows

Region *
🔔 Not finding your App Service Plan? Try a different region or select your App Service Environment.

Pricing plans

App Service plan pricing tier determines the location, features, cost and compute resources associated with your app. [Learn more](#) [🔗](#)

Windows Plan (East US) * ⓘ
[Create new](#)

Pricing plan
[Explore pricing plans](#)

Zone redundancy

An App Service plan can be deployed as a zone redundant service in the regions that support it. This is a deployment time only decision. You can't make an App Service plan zone redundant after it has been deployed. [Learn more](#) [🔗](#)

Zone redundancy ☐ **Enabled:** Your App Service plan and the apps in it will be zone redundant. The minimum App Service plan instance count will be three.
☒ **Disabled:** Your App Service Plan and the apps in it will not be zone redundant. The minimum App Service plan instance count will be one.

Figure 40: Create new web app

Step 2: Select method publish

Publish

Where are you publishing today?

Target

- Azure**
Publish your application to the Microsoft cloud
- Docker Container Registry**
Publish your application to any supported Container Registry that works with Docker images
- Folder**
Publish your application to a local folder or file share
- FTP/FTPS Server**
Publish your application to an FTP/FTPS server
- Web Server (IIS)**
Publish your application to IIS using Web Deploy or Web Deploy Package
- Import Profile**
Import your publish settings to deploy your app

Back Next Finish Cancel

Figure 41: Select method publish

Step 3: Select Azure app service

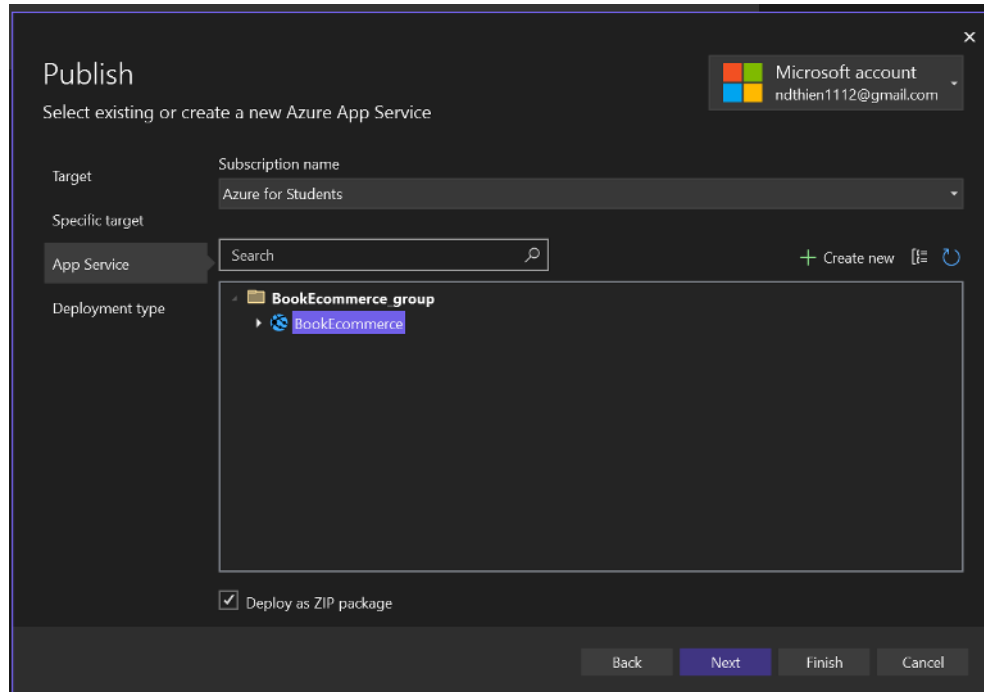


Figure 42: Select existing service

Result:

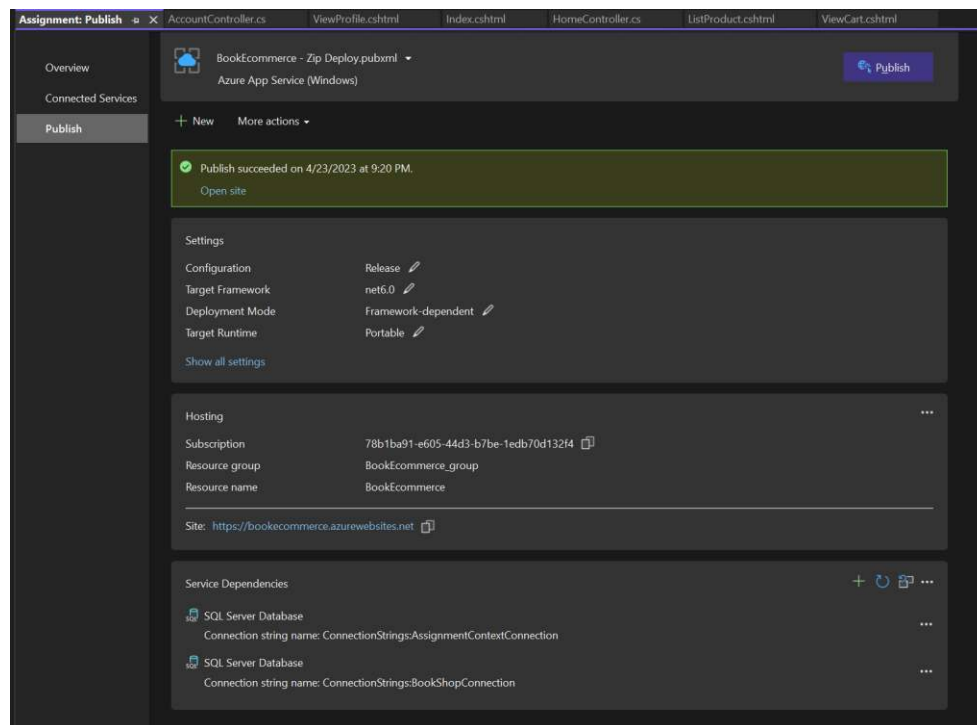


Figure 43: Result

Azure SQL Database
Create new

Microsoft account
ndthien1112@gmail.com

Database name
Assignment_db

Subscription name
Azure for Students

Resource group
BookEcommerce_group (East Asia) New...

Database server
assignmentdbserver* (East Asia) New...

Database administrator username (must have permissions to create)
adminbookshop

Database administrator password
.....

Creating: Azure SQL Database... Create Cancel

IV/ APPLICATION EVALUATION (P6)

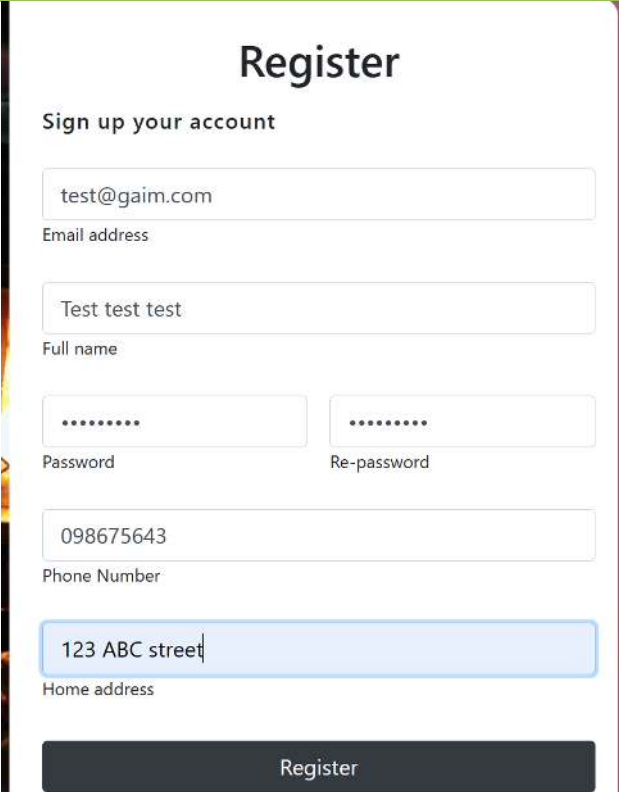
1. Review the performance of the application

1.1 Test case

No	Name of test case	Description of test case
1	Register an account	After users input their information into the corresponding fields and click on the register button, the system will add a new row to the table user in the database.

2	Login by an account	After users input their information into the corresponding fields and click on the login button, they can access the system.
3	Search a book by name	After inputting a name of a book into the search field, the system will display all books whose name is similar to the user's entered data.
4	Search books by category	After selecting a category and clicking on the filter button, users will view all books whose category is equal to customers' desired category.
5	Add books to the cart	After clicking on add button, the book will exist in the cart.
6	Increase the quantity	After clicking on the increase button, the quantity of the book will increase in the cart
7	Remove a book from the cart	After clicking on the remove button, the book will be removed from the cart
8	Clear the cart	After clicking on the clear button, all books in the cart will be removed
9	Edit profile	Inputting new information and clicking on the save button, the information of the customer will be changed.
10	Request a new category	After owners request a new category, the admin will see it.
11	Add a new book	Owners can add a new book by inputting the corresponding information into the fields.
12	Edit a book	Owners can edit a book by inputting the corresponding information into the fields.
13	Delete a book	Owners can remove a book by clicking on the remove button.
14	View information on accounts	Admin can view information about all accounts in the system.
15	Change the password of an account	Admin can change the password for any account by inputting the new password and clicking on the save button

1.2 Test result

Test case	Test Name	Expected result	Actual result	Status
1	Register an account	After users click on the register button, a new record of table users is added	 <p>The screenshot shows a 'Register' form with the following fields and values:</p> <ul style="list-style-type: none"> Sign up your account (header) Email address: test@gaim.com Full name: Test test test Password: (masked) Re-password: (masked) Phone Number: 098675643 Home address: 123 ABC street Register button: A dark grey button labeled 'Register'. 	Pass

100 %

Results Messages

	Id	FullName	ProfilePicture	Location	UserName	Normalized UserName	Email
1	43146305-7334-4501-9fe1-5aeb1b2a63d1	Test test test	NULL	123 ABC street	Testtesttest	TESTTESTTEST	test@gmail.com
2	71430978-55e7-4410-8dd0-62da25bf096a	Huy Tran	NULL	123 ABC street, Ha Noi, Viet Nam	HuyTran	HUYTRAN	customer1@gmail.com
3	abcd	123456789	abc	123 ABC street	NULL	NULL	betngao@gmail.com
4	c9cb0388-bd8e-4709-a6a2-532ed2512591	Thien Nguyen Duc	48520ae6-1cf2-4d79-8563-f6b1507aa6a4.png	123 ABC street	ThienNguyenDuc	THIENNGUYENDUC	betngado@gmail.com
5	eea4514e-a065-4da1-9a47-bf0942af2682	shop owner	NULL	123 ABS street	shopowner	SHOPOWNER	betngadoddd@gmail.com

2

Login into the system

After clicking on the login button, users will see their name on the web page

Blog Featured Products wishlist

Hello Testtesttest Profile Logout Shop Management English

WPFStore Home Shop New Book About us

All Departments
Total: 10 Products

Search for products Search

Shop Status

Pass

3	Search a book by name	The system displays customers' desired books after searching		Pass
4	Filter books by a category	The system displays customers' desired books after filtering		Pass

5

Add a book to the cart

A book will exist in the cart after clicking on the add button

Shopping Cart

1 items



Lauren McBrayer
Like a House on Fire

- 1 +

\$16

×

← Back to shop

Clear Cart

Pass

6

Increase the quantity of a book in the cart

The quantity will increase after clicking on the "+" button

Shopping Cart

1 items



Lauren McBrayer
Like a House on Fire

- 2 +


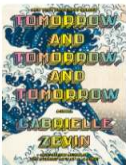
\$32

×

← Back to shop

Clear Cart

Pass

7	Remove a book from the cart	The book will be removed after clicking on the remove button	<div> <h2>Shopping Cart</h2> <p>1 items</p> <hr/> <p>Your cart is empty.</p> <hr/> </div>	Pass
8	Clear the cart	All books will be removed from the cart after clicking on the clear button	<div> <h2>Shopping Cart</h2> <p>1 items</p> <p>Before clear</p> <hr/> <div>  <div> <p>Tatsuki Fujimoto</p> <p>Chainsaw Man, Vol. 11</p> </div> <div> <p>−</p> <p>2</p> <p>+</p> </div> <div> <p>\$32</p> <p>×</p> </div> </div> <hr/> <div>  <div> <p>Christopher Buehlman</p> <p>Tomorrow, and Tomorrow, and Tomorrow</p> </div> <div> <p>−</p> <p>1</p> <p>+</p> </div> <div> <p>\$22</p> <p>×</p> </div> </div> <hr/> </div>	Pass

Shopping Cart

After clear

Your cart is empty.

9

Edit profile

Information of
an account will
be changed
after editing



Profile Detail

Full name

Test test test

Home address

123 ABC street

Email

test@gmail.com


Phone Number

098675643

Edit Profile

Pass

After edit



Profile Detail

Full name
After editing

Home address
After editing

Email
afteredit@gmail.com

Phone Number
098675643

Edit Profile

10

Request a
new category

A new category
will appear on
the admin page
after owners
send a request

Category Table

ID	Category Name	Description	Status	Action
7	drama	for book that describes many dramatical stories	Processing	Accept Reject

Page 1 of 1

1

Pass

11

Add a new book

A new book will appear after clicking on the add button

← ↻ 📁 https://localhost:7291/shop/product/addproduct

Purple ☰ 🔍 Search projects

Add A New Product

Input the information below

Name Book

Spy X Family, Vol. 9 (Testing)

Name cannot be empty !!

Price

12

DiscountPrice

10

Quantity

23

Quantity must be between 1 and 1000

Author

Tatsuki Fujimoto

Author cannot be empty !!

Pass

--Sorting by Status--
--Sorting by Price--

Book Table

ID	Product Name	Image	Price	Quantity	Status	Action
11	Let This Radicalize You: Organizing and Revelution		18.0000	22	InStock	<input type="button" value="Edit"/> <input type="button" value="Remove"/>
12	Spy X Family, Vol. 9		9.0000	22	InStock	<input type="button" value="Edit"/> <input type="button" value="Remove"/>
13	Spy X Family, Vol. 9 (Testing)		10.0000	23	InStock	<input type="button" value="Edit"/> <input type="button" value="Remove"/>

Page 3 of 3

12

Edit a book

Information about a book will be changed after owners edit it.

--Sorting by Status--
--Sorting by Price--

Book Table

ID	Product Name	Image	Price	Quantity	Status
11	Let This Radicalize You: Organizing and Revelution		18.0000	22	InStock
12	Spy X Family, Vol. 9		9.0000	22	InStock
13	Spy X Family, Vol. 9 (After edit)		10.0000	23	InStock

Page 3 of 3

Pass

13

Remove a book

After clicking on the remove button, then continue to click on the ok button, the book will be removed

The screenshot shows a web application interface for a bookstore. At the top, there's a navigation bar with the 'Purple' logo and a search bar. Below the navigation bar, there's a sidebar with 'Dashboard' and 'Category' links. The main content area displays a 'Book Table' with columns: ID, Product Name, Image, Price, Quantity, Status, and Action. Two books are listed: 'Let This Radicalize You: Organizing and Revelation' (ID 11, Price 18.0000, Quantity 22, Status 'InStock') and 'Spy X Family, Vol. 9' (ID 12, Price 9.0000, Quantity 22, Status 'InStock'). Each book has 'Edit' and 'Remove' buttons. A confirmation dialog box is open over the 'Remove' button of the first book, asking 'Are you sure you want to remove?' with 'OK' and 'Cancel' buttons. The URL in the browser is 'https://localhost:7291/shop/product?page=3'.

Pass

14

View information on accounts

Admin can view information of account when he logs in the system

The screenshot shows a web application interface for managing user accounts. The title is 'Account Of Owner'. Below the title, there's a table with columns: ID, Name Customer, Email, Phone Number, and Action. Three accounts are listed: 'After editing' (ID 43146305-7334-4501-9fe1-5aeb1b2a63d1, Email afteredit@gmail.com, Phone Number 098675643), 'Huy Tran' (ID 71430978-55e7-4410-8dd0-62da25bf096a, Email customer1@gmail.com, Phone Number 0123123123), and 'shop owner' (ID eea4514e-a065-4da1-9a47-bf0942af2682, Email betngadodd@gmail.com, Phone Number 1234912312). Each account has 'Change Password' and 'Remove' buttons. The page number 'Page 1 of 1' is shown at the bottom.

Pass

15

Reset
password

After change
password,
users can log in
with the new
password

Update password for email: betngado@gmail.com

Update below:

.....

Password

.....

Re-password

Submit

Cancel

Pass

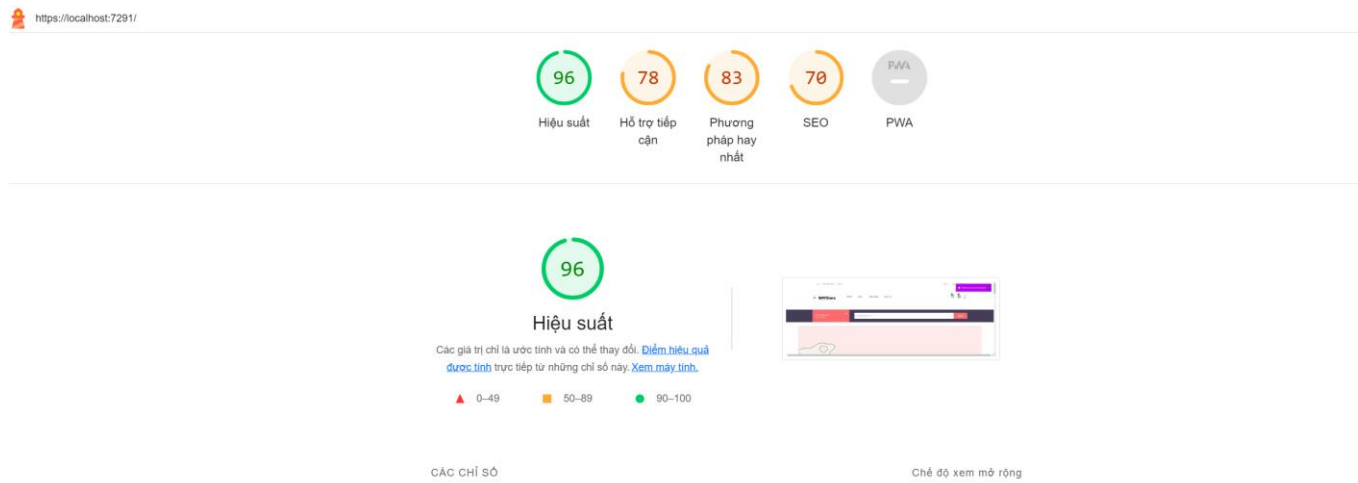


Figure 44: Measure Performance

Overall, the results are quite satisfactory as my site's performance is up to 96%. However, other indicators are only approximately 80% and this is really not optimal for an online book selling system. Therefore, we will provide a new update to improve the system as quickly as possible in the near future.

2. Conclude whether the application adapts all requirements or it needs to be improved later

Based on the evaluation of the specified requirements around web pages, entity models, controllers, application features, and use of HTML/CSS, it can be concluded that the application satisfies all the mandatory requirements and adapts most of the desired requirements.

The application has 18 web pages (views) covering all the necessary user flows and features. It has 7 independent entity models including Book, Category, Order, OrderDetail, User, Role, and UserRole to store data efficiently. There are 9 controllers implemented to handle requests and responses for all roles. While the application applies HTML5 and CSS3 as intended to structure the content and styling, the CSS3 stylesheets can be optimized further. Overall, the application achieves all the mandatory requirements around web pages, entity models, and controllers as desired. But CSS stylesheets require further improvement to fully adapt the specifications. With these upgrades, the application can reach its maximum potential.

Besides, Bootstrap was used to boost the process of building the interface for the application and to ensure that the design of the website is consistent and responsive across different devices and screen sizes.

In terms of functionality required by our customers, our system covers them all:

- There is a Home Screen that will be the default screen displayed upon processing the user login and the user log-out options.
- There is a List Products Screen that will display all books that the customer can buy.
- There is a Product Detail Screen that will help to display detailed information about a specific book.
- There is a Profile Screen that will be used for displaying his/her profile.
- There is a Register Screen that will be used for customer registration.
- There is a Book Screen that will be used for adding, updating, searching, or deleting a product. Each book must have a specific Category.
- There is a Category Request Screen that will be used to make a request to the admin for adding the new book category if it does not exist.
- There is a Record Screen that will be used to display all records ordered by the customer.
- There is a Customer Screen and a Store Owner screen that will be used to display the account information of all Customers. It will be able to reset the password of the Customer if needed.
- There is a Category Approval Screen that will allow the admin to approve or reject the new book category request made by the Store Owner.

Overall, our application met all requirements of customers, but those requirements are not enough to optimize the process of business in an e-commerce bookstore. Therefore, if we have more time and opportunities, we're sure to release a new update with more functions making the management of the owners' process become more convenient and easier.

3. Analyze the factors that influence the performance of the application

There are many factors that can influence the performance of an e-commerce website, including the quality of the code used to build it. Generally, some essential factors below should be examined and studied to make an application have a better performance:

- **Website design:** The design of the website can have a significant impact on its performance. A well-designed website that is easy to navigate and visually appealing can encourage users to spend more time on the site and make more purchases.
- **Website speed:** The speed of the website is crucial to its success. Slow loading times can lead to frustration and cause users to abandon the site. Optimizing images, using caching techniques, and minimizing the use of external resources can help improve website speed.
- **Server performance:** The performance of the server hosting the website is also critical. A slow or overloaded server can cause the site to crash or become unresponsive, leading to a poor user experience.
- **Mobile optimization:** With more users accessing e-commerce sites on mobile devices, it is essential to ensure that the site is optimized for mobile use. This includes using responsive design techniques and optimizing images and other resources for smaller screens.
- **Code quality:** The quality of the code used to build the website can have a significant impact on its performance. Well-written, optimized code can improve website speed and reduce the risk of crashes or other technical issues.
- **User experience:** The overall user experience is critical to the success of an e-commerce website. This includes factors such as website design, ease of use, product information, and customer service. Specifically, to ensure that visitors enjoy their experience, make sure the website has a clean and attractive interface. Make sure that each page is organized logically and that the user doesn't get lost while navigating through the site.

4. Evaluate the strengths and weaknesses of the application

Strengths:

One of the biggest strengths of this application is its user-friendly interface. The interface is designed in such a way that it is easy to navigate, and users can quickly find what they are looking for. Additionally, the application is equipped with various functions such as adding, editing, deleting, searching, and sorting, which makes it easy for users to manage their book inventory.

Another strength of this application is that it caters to the needs of all three user types, i.e., customer, shop, and admin. Each user type has its own set of features, and the application fully meets the requirements of each user type. For instance, customers can browse through the book catalog, add books to their cart, and make purchases, while shops can manage their book inventory, and admins can manage the overall functioning of the application.

Weaknesses:

One weakness of this application is that it still has some responsive errors. This means that the website may not display correctly on all devices, and users may face difficulties accessing it. This can be a major issue as it can lead to a decline in user engagement and result in a loss of potential customers.

Another weakness of this application is that it may not be able to handle a large number of visits. This means that if a large number of users try to access the website simultaneously, it may crash or become slow, leading to a poor user experience. This can be a significant drawback, particularly during peak times such as book launches or sales.

Lastly, the application is not entirely secure in terms of security. This can be a significant concern as users may not feel comfortable sharing their personal and financial information on a website that is not secure. This could lead to a loss of customer trust, which could have long-term negative effects on the success of the application.